IBM eServer pSeries 690 Turbo c/s Model 7040-681

using

Oracle Database 10g Enterprise Edition

IBM TXSeries Developer for AIX V5

TPC Benchmark™ C

Full Disclosure Report

Second Edition

October 17, 2003

Special Notices

The following terms used in this publication are trademarks of **International Business Machines** Corporation in the United States and/or other countries:

eServer pSeries
AIX
IBM

The following terms used in this publication are trademarks of other companies as follows:

TPC Benchmark	Trademark of the Transaction Processing Performance Council
ORACLE, SQL*Loader	Trademark of Oracle, Inc.
Oracle10g, SQL*Net and SQL*Plus	Trademark of Oracle, Inc.

Second Edition October 17, 2003

The information contained in this document is distributed on an AS IS basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used.

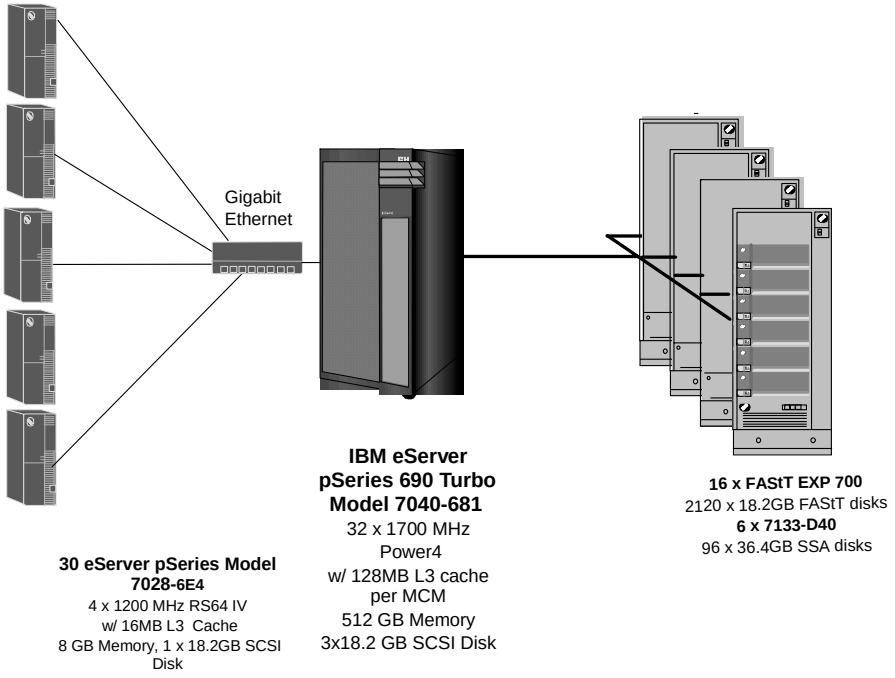
It is possible that this material may contain references to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such products, programming, or services in your country.

All performance data contained in this publication was obtained in a controlled environment, and therefore the results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data in their specific environment.

Request for additional copies of this document should be sent to the following address:

TPC Benchmark Administrator
IBM Commercial Performance
Mail Stop 9571
11501 Burnet Road
Austin, TX 78758
FAX Number (512) 838-1852

IBM / ORACLE	IBM eServer pSeries 690 Turbo Model 7040-681		TPC-C Rev. 5.1	
			Report Date Sept 12, 2003	
Total System Cost	**TPC-C Throughput**	**Price/Performance**	**Availability Date**	
$6,574,014	**768,839.40tpmC**	**8.55$/tpmC**	**February 29, 2004**	
Processors	**Database Manager**	**Operating System**	**Other Software**	**No. Users**
32 x database 120 x clients	**Oracle Database 10g Enterprise Edition**	**AIX 5L V5.2**	**IBM TXSeries Developer for AIX V5**	**607,680**



System Components	Clients		Server	
	Quantity	Description	Quantity	Description
Processor	**30**	**4 x1200 MHz POWER4 W/16MB L3 cache each**	**1**	**32 x 1700 MHz POWER4 w/ 128 MB L3 cache per MCM, 4 MCM in the SUT**
Memory	**8 GB**		**512 GB**	
Disk Controllers	**1**	**SCSI-2 (Integrated)**	**1 6 66**	**SCSI-2 (Integrated) SSA Adapters Fibre Channel Adapters**
Disk Drives	**1**	**18.2 GB SCSI each client**	**2,120 96 3**	**18.2 GB FAStT disk 36.4 GB SSA disk 18 .2 GB SCSI disk**
Total Storage	**16.93 GB each client**		**37,325.49**	
Terminals	**1**	**System Console**	**1**	**System Console**

Description	Part No.	Source	Unit Price	Qty	Ext Price	Maint Price
Server Hardware						
IBM eServer pSeries 690 (CD, cables, clock card)	7040-681	1	13,394	1	13,394	20,760
128MB (4x32) L3 Cache, 1700MHz 8w MCM	4199	1	375,000	4	1,500,000	372,000
64GB H-Memory (inward and outward 4488,4489)	4488	1	344,000	4	1,376,000	
Std CEC Fan Pwr cblgrp, service proc. 6161, 6162	6161	1	600	1	600	
Standard CEC DC/DC Converter Assembly	6170	1	4,700	2	9,400	
Power Cable Group, 1 thru 4 proc. module	6181	1	720	1	720	
Additional DC/DC Converter Assembly, (DCA)	6189	1	4,200	3	12,600	
Capacitor book (for 2 MCMs)	6198	1	1,800	2	3,600	
Service Processor + Dual Rio Loop (GX), I/O loop adapter	6418	1	15,000	1	15,000	
Backplane, Central Electronics Complex	6565	1	50,000	1	50,000	
Attachment Cable, Hardware Management Console	8121	1	75	1	75	
Media Drwr, 1U, Op Pnl/Dskt/4 SCSI Media Bays	8692	1	1,700	1	1,700	
I/O Drawer	7040-61D	1	3,980	8	31,840	32,640
Converter Cable, VHDCI to P, 0.3M	2118	1	60	8	480	
SCSI cable - B&C to Media Drawer	2122	1	275	1	275	
RIO-G Cable, 500Hz, 0.5m, 2*2m (3145, 3149)	3145	1	1010	8	8,080	
36.4 GB 10,000 RPM Ultra3 SCSI Disk Drive	3158	1	1,283	3	3,849	
IBM Gigabit Ethernet-SX PCI-X Adapter	5700	1	1,700	1	1,700	
Cable Grp, 4xUPIC/2xRIO, BPM(IO#1 thru 4) (6121-6129)	6121	1	3,200	1	3,200	
I/O Drwr DC/DC Converter assembly, (DCA)	6172	1	4,000	16	64,000	
Power Cable, B&C to Media Drawer 6179,SCSI adapter 6201	6179	1	695	1	695	
Advanced Serial RAID Adapter, 32MB Cache (6230, 6235)	6230	1	3,575	6	21,450	
2 Gigabit Fibre Channel PCI-X Adapter	6239	1	2,582	66	170,412	
Ultra3 SCSI 4-Pack	6564	1	500	3	1,500	
B&C Planar, 10 PCI Slots, 2x integrated SCSI	6571	1	8,000	16	128,000	
Rack, Front doors, Back doors	7040-61R	1	10,550	1	10,550	3,960
Bulk Power Regulator (BPR)	6186	1	4,000	4	16,000	
Bulk Power Controller (BPC), 4 Fans + 3 DCAs	6187	1	1,900	2	3,800	
Bulk Power Distribution (BPD), 10 DCAs	6188	1	3,500	4	14,000	
Line Cord, 60A, 14', IEC309 Plug, Chargeable	8678	1	1,000	2	2,000	
Regatta-H Bulk Power Module (2x BPA, EPO PNL)	8690	1	5,000	1	5,000	
Expansion Rack 24", 42 EIA	8691	1	8,000	1	8,000	
System Console, serial attach. cable 6M	7316	1	4,400	1	4,400	
IBM T541H 15 inches TFT Color Monitor, keyboard, graphics	3637	1	974	1	974	
				Subtotal	**3,483,294**	**429,360**
Storage						
System Rack Model T42 (6081,6098)	7014-T42	1	4,270	1	4,270	888
Additional Power Distribution Unit	6171	1	1,000	1	1,000	
				Subtotal	**5,270**	**888**
SSA Disk Subsystem,50/60 Hz AC, pwr supply, cover, cabl	7133-D40	1	15,125	6	90,750	26,496
Advanced 10K rpm/36.4 GB Disk Module	8536	1	5,900	96	566,400	
				Subtotal	**657,150**	**26,496**
Racked FAStT Storage Solution	2101-200	1	4,850	16	77,600	12,288
FAStT EXP 700	1740-1RU	1	6,000	160	960,000	230,400
Short Wave SFP	2210	1	499	320	159,680	
18.2 GB 15K drive	5211	1	681	2,120	1,443,720	
FAStT700 Storage Server	1742-1RU	1	46,499	32	1,487,968	46,080
Short Wave SFP GBIC	2210	1	499	128	63,872	
				Subtotal	**4,192,840**	**288,768**
Server Software						
AIX 5L v5.2 (media only)	5765-E61	1	50	1	50	
AIX SWMA Subsription	5771-SUB	1	247	96	23,712	
AIX SWMA Support	5771-SPT	1	697	96	0	66,912
IBMC for AIX V5	D5A1ELL	1	559	1	559	104
Oracle Database 10g Enterprise Edition, Per Processor, Unlimited Users, for 3 years		2	20,000	32	640,000	
Oracle Database Server Support Package for 3 years		2				6,000
				Subtotal	**664,321**	**73,016**
Client Hardware and Software						
pSeries 630-6E4, CD-ROM, 18.2 GB Disk, Pwr Supply	7028-6E4	1	3,941	30	118,230	90,000
FC I Exp Config (420C/420E) AIX	8037	1	23,317	30	699,510	
T541H 15" TFT Color Monitor, Captured Cable, keyboard	3637	1	562	1	562	
POWER GXT135P Graphics Accelerator -Digital support	2849	1	412	1	412	
SCSI Connector Cable + 4-Pack How Swap Back Plane	4254	1	354	30	10,620	
2048MB (4x512MB) SDRAM DIMM Memory,Express	8080	1	0	120	0	
4-port 10/100 Mbps Ethernet adapter	4961	1	1,250	90	112,500	
2-way 1.2 GHz POWER4 processor card, Express	8107	1	0	60	0	
AIX SWMA Subsription	5773-SUB	1	157	240	37,680	
AIX SWMA Support	5773-SPT	1	427	240	0	102,480
AIX 5L version 5	5692-A5L	1	50	30	1,500	
TXSeries Developer for AIX V5, per proc, w/3 yrs maint.		1	2,599	120	311,904	
				Subtotal	**1,292,918**	**192,480**
Third Party Hardware						
WideBand Series Fast Ethernet/Gigabit Switch	WB24T2EML3	3	1,449	4	5,796	
				Subtotal	**5,796**	
Oracle Mandatory E-Business Discount on License and Support		2	(129,200)	1	(129,200)	
				Discounts	**(4,211,211)**	**(398,172)**
				Total	**5,961,178**	**612,836**

Audited by: Francois Raab, InfoSizing (www.infosizing.com)

Pricing Sources:

1=IBM: Bill Casey, eServer pSeries Offering Manager, wrcasey@us.ibm.com, 512-838-1422

2=Oracle: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 650-506-2118

3=WideBand Corp.

Three-Year Cost of Ownership	**$6,574,014**
tpmC	**768,839.40**
$/tpmC	**8.55**

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 terms, please inform the TPC at pricing@tpc.org. Thank you

Numerical Quantities Summary for the IBM eServer pSeries 690 Turbo Model 7040-681

MQTH, computed Maximum Qualified Throughput: 768,839.40**tpmC**

Response Times (in seconds)	90th %	Average	Maximum
New Order	0.64	0.31	25.31
Payment	0.63	0.30	24.46
Order-Status	0.63	0.30	21.96
Delivery (interactive)	0.19	0.11	21.00
Delivery (deferred)	0.05	0.03	4.94
Stock-Level	0.63	0.30	23.03
Menu	0.01	0.01	2.53

Transaction Mix, in percent of total transactions	Percent
New Order	44.90%
Payment	43.02%
Order-Status	4.01%
Delivery	4.01%
Stock-Level	4.02%

Keying/Think Times (in seconds)	Min.	Average	Max.
New Order	18.00/0.01	18.01/12.02	18.11/120.20
Payment	3.00/0.01	3.01/12.02	3.12/120.20
Order-Status	2.00/0.01	2.01/10.01	2.09/100.10
Delivery	2.00/0.01	2.01/5.02	2.09/50.20
Stock-Level	2.00/0.01	2.01/5.02	2.10/50.20

Test Duration	
Ramp-up Time	1 hour 9 min 47 sec
Measurement interval	2 hours
Transactions during measurement interval (all types)	304667487
Ramp-down time	17 minutes

Checkpointing	
Number of checkpoints	4
Checkpoint interval	29 min 1 sec

Abstract

This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification Revision 5.1 dated December 2002 for measurements on the IBM eServer pSeries 690 Turbo Model 7040-681. The software used on the IBM eServer pSeries 690 Turbo Model 7040-681 includes AIX 5L Version 5.2 operating system, Oracle 10g Server database manager, and IBM TXSeries Developer for AIX V5 for transaction manager.

IBM eServer pSeries 690 Turbo Model 7040-681

Company Name	System Name	Data Base Software	Operating System Software
IBM Corporation Oracle Corporation	IBM eServer pSeries 690 Turbo Model 7040-681	Oracle Database 10g Enterprise Edition	AIX 5L Version 5.2

Total System Cost	TPC-C Throughput	Price/Performance
• Hardware • Software • 3 Years Maintenance	Sustained maximum throughput of system running TPC-C expressed in transactions per minute	Total system cost/tpmC
$6,574,014	**768,839.40tpmC**	**8.55$/tpmC**

Preface

TPC Benchmark™ C Standard Specification was developed by the Transaction Processing Performance Council (TPC). It was released on August 13, 1992 and updated with revision 5.1 on December 2002.

This is the full disclosure report for benchmark testing of the IBM eServer pSeries 690 Turbo Model 7040-681 according to the TPC Benchmark™ C Standard Specification.

TPC Benchmark™ C exercises the system components necessary to perform tasks associated with that class of on-line transaction processing (OLTP) environments emphasizing a mixture of read-only and update intensive transactions. This is a complex OLTP application environment exercising a breadth of system components associated by such environments 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
- Data bases consisting of many tables with a wide variety of sizes, attributes, and relationships
- Contention on data access and update

This benchmark defines four on-line transactions and one deferred transaction, intended to emulate functions that are common to many OLTP applications. However, this benchmark does not reflect the entire range of OLTP requirements. The extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

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

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.

1. General Items

1.1 Application Code Disclosure

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 and output functions.

Appendix A contains the RS/6000 application code for the five TPC Benchmark™ C transactions. Appendix D contains the terminal functions and layouts.

1.2 Benchmark Sponsor

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

This benchmark was sponsored by **International Business Machines Corporation** and **Oracle Corporation.**

1.3 Parameter Settings

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

- *Data Base tuning options*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and application configuration parameters.*

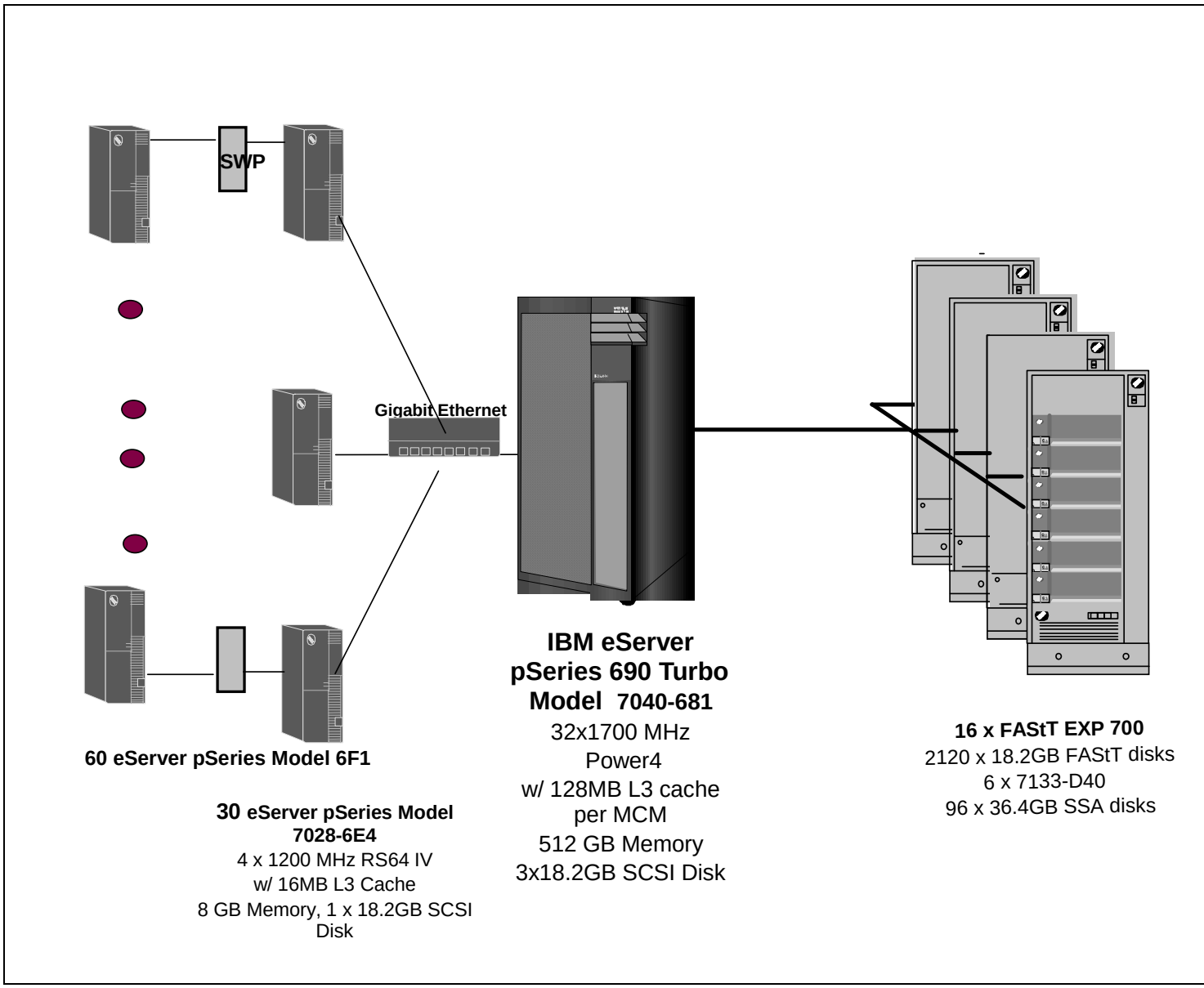
Appendix B contains the system, data base, and application parameters changed from their default values used in these TPC Benchmark™ C tests.

1.4 Configuration Diagrams

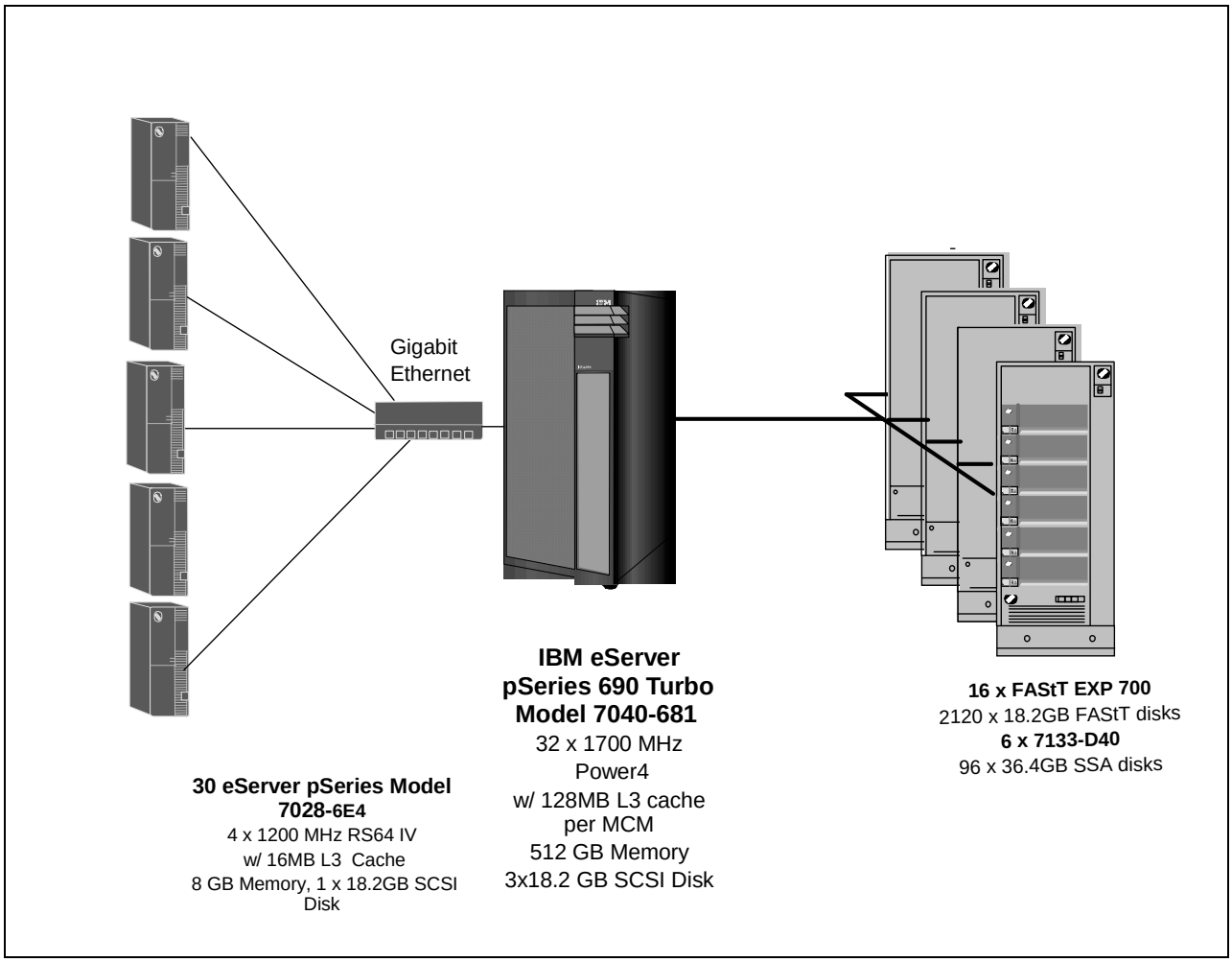
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- *Number and type of processors*
- *Size of allocated memory, and any specific mapping/partitioning of memory unique to the test*
- *Number and type of disk units (and controllers, if applicable)*
- *Number of channels or bus connections to disk units, including the protocol type*
- *Number of LAN (e.g. Ethernet) connections, including routers, work stations, terminals, etc, that were physically used in the test or are incorporated into the pricing structure (see Clause 8.1.8)*
- *Type and run-time execution location of software components (e.g. DBMS, client processes, transaction monitors, software drivers, etc)*

IBM eServer pSeries 690 Turbo Model 7040-681 Benchmark Configuration



IBM eServer pSeries 690 Turbo Model 7040-681 Priced Configuration



2. Clause 1: Logical Data Base Design Related Items

2.1 Table Definitions

Listings must be provided for all table definition statements and all other statements used to setup the data base.

Appendix C contains the table definitions and the database load programs used to build the data base.

2.2 Database Organization

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

Physical space was allocated to Oracle 10g Server on the server disks according to the details provided in Appendix C. The size of the space segments on each disk was calculated to provide even distribution of data across the disk subsystem.

2.3 Insert and/or Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restriction in the SUT data base 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.

There were no restrictions on insert and/or delete operations to any of the tables. The space required for an additional five percent of the initial table cardinality was allocated to Oracle 10g Server and priced as static space.

2.4 Horizontal or Vertical Partitioning

While there are 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.

3. Clause 2: Transaction and Terminal Profiles Related Items

3.1 Verification for the Random Number Generator

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

The srandom(), getpid() and gettimeofday() functions are used to produce unique random seeds for each driver. The drivers use these seeds to seed the srand(), srandom() and srand48() functions. Random numbers are produced using wrappers around the standard system random number generators.

The negative exponential distribution uses the following function to generate the distribution. This function has the property of producing a negative exponential curve with a specified average and a maximum value 4 times the average.

```
        const double RANDOM_4_Z = 0.89837799236185
        const double RANDOM_4_K = 0.97249842407114

        double neg_exp_4(double average {
                return - average * (1/RANDOM_4_Z * log (1 - RANDOM_4_K * drand48())));
}
```

The random functions used by the driver system and the data base generation program were verified. The C_LAST column was queried to verify the random values produced by the database generation program. After a measurement, the HISTORY, ORDER, and ORDER_LINE tables were queried to verify the randomness of values generated by the driver. The rows were counted and grouped by customer and item numbers.

Here is an example of one SQL query used to verify the random number generation functions:

- create table TEMP (W_ID int, D_ID, C_LAST char(16), CNTR int);
- insert into TEMP select C_W_ID, C_D_ID, C_LAST, COUNT(*) from CUSTOMER group by C_W_ID, C_D_ID, C_LAST;
- select CNTR, COUNT(*) from TEMP group by CNTR order by 1;

3.2 Input/Output Screens

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

The screen layouts corresponds exactly to the layout corresponding in clauses 2.4.3, 2.5.3, 2.6.3, 2.7.3 and 2.8.3 of the TPC-C specifications.

3.3 Priced Terminal Features

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 emulated workstations, IBM RS/6000 Model 44P-170, are commercially available and support all of the requirements in Clause 2.2.2.4.

3.4 Presentation Managers

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

The RS/6000 Model 44P-170 workstations did not involve screen presentations, message bundling or local storage of TPC-C rows. All screen processing was handled by the client system. All data manipulation was handled by the server system.

3.5 Home and Remote Order-lines

The percentage of home and remote order-lines in the New-Order transactions must be disclosed.

Table 3-1 show the percentage of home and remote transactions that occurred during the measurement period for the New-Order transactions.

3.6 New-Order Rollback Transactions

The percentage of New-Order transactions that were rolled back as a result of an illegal item number must be disclosed.

Table 3-1 show the percentage of New-Order transactions that were rolled back due to an illegal item being entered.

3.7 Number of Items per Order

The number of items per order entered by New-Order transactions must be disclosed.

Table 3-1 show the average number of items ordered per New-Order transaction.

3.8 Home and Remote Payment Transactions

The percentage of home and remote Payment transactions must be disclosed.

Table 3-1 show the percentage of home and remote transactions that occurred during the measurement period for the Payment transactions.

3.9 Non-Primary Key Transactions

The percentage of Payment and Order-Status transactions that used non-primary key (C_LAST) access to the data base must be disclosed.

Table 3-1 show the percentage of non-primary key accesses to the data base by the Payment and Order-Status transactions.

3.10 Skipped Delivery Transactions

The percentage of Delivery transactions that were skipped as a result of an insufficient number of rows in the NEW-ORDER table must be disclosed.

Table 3-1 show the percentage of Delivery transactions missed due to a shortage of supply of rows in the NEW-ORDER table.

3.11 Mix of Transaction Types

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

Table 3-1 show the mix percentage for each of the transaction types executed by the SUT.

3.12 Queueing Mechanism of Delivery

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

The Delivery transaction was submitted using an RPC call to an IBM TXSeries Developer for AIX V5, Encina interface transaction manager (TM). Encina returns an immediate response to the calling program and schedules the work to be performed. This allows the Delivery transaction to be submitted, obtain an interactive response and queue the actual data base transaction for deferred execution. Please see the application code in Appendix A for details.

Table 3-1 Numerical Quantities for Transaction and Terminal Profiles

New Order	IBM eServer pSeries 690 Turbo Model 7040-681
Percentage of Home order lines	99.0%
Percentage of Remote order lines	1.00%
Percentage of Rolled Back Transactions	1.00%
Average Number of Items per order	10
Payment	
Percentage of Home transactions	85.00%
Percentage of Remote transactions	15.00%
Non-Primary Key Access	
Percentage of Payment using C_LAST	59.99%
Percentage of Order-Status using C_LAST	59.98%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.90%
Payment	43.02%
Order-Status	4.01%
Delivery	4.01%
Stock-Level	4.02%

4. Clause 3: Transaction and System Properties

The results of the ACID test must be disclosed along with a description of how the ACID requirements were met.

All ACID tests were conducted according to specification.

4.1 Atomicity Requirements

The system under test must guarantee that data base 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.

4.1.1 Atomicity of Completed Transaction

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

The following steps were performed to verify the Atomicity of completed transactions.

1. The balance was retrieved from the CUSTOMER table for a random Customer, District and Warehouse giving BALANCE_1.
2. The Payment transaction was executed for the Customer, District and Warehouse used in step 1.
3. The balance was retrieved again for the Customer used in step 1 and step 2 giving BALANCE_2. It was verified that BALANCE_1 was greater than BALANCE_2 by AMT.

4.1.2 Atomicity of Aborted Transactions

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

The following steps were performed to verify the Atomicity of the aborted Payment transaction:

1. The Payment application code was changed to execute a rollback of the transaction instead of performing the commit.
2. Using the balance, BALANCE_2, from the CUSTOMER table retrieved for the completed transaction, the Payment transaction was executed for the Customer, District, and Warehouse used in step 1 of the section 4.1.1, using a payment amount (AMT) of 410.00. The transaction rolled back due to the change in the application code from step 1.
3. The balance was retrieved again for the Customer used for step 2 giving BALANCE_3. It was verified that BALANCE_2 was equal to BALANCE_3.

4.2 Consistency Requirements

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

Verify that the data base is initially consistent by verifying that it meets the consistency conditions defined in Clauses 3.3.2.1 to 3.3.2.4. Describe the steps used to do this in sufficient detail so that the steps are independently repeatable.

4.2.1 Consistency Condition 1

Entries in the WAREHOUSE and DISTRICT tables must satisfy the relationship:

- *W_YTD = sum(D_YTD)*

for each warehouse defined by (W_ID = D_W_ID)

4.2.2 Consistency Condition 2

Entries in the DISTRICT, ORDER, and NEW-ORDER tables must satisfy the relationship:

- *D_NEXT_O_ID - 1 = max(O_ID) = max(NO_O_ID)*

for each district defined by (D_W_ID = O_W_ID = NO_W_ID) and (D_ID = O_D_ID = NO_D_ID). This condition does not apply to the NEW-ORDER table for any districts which have no outstanding new orders.

4.2.3 Consistency Condition 3

Entries in the New-Order table must satisfy the relationship:

- *max(NO_O_ID) - min(NO_O_ID) + 1 = [number of rows in the New-Order table for this district]*

for each district defined by NO_W_ID and NO_D_ID. This condition does not apply to any districts which have no outstanding new orders.

4.2.4 Consistency Condition 4

Entries in the ORDER and ORDER-LINE tables must satisfy the relationship:

- *sum(O_OL_CNT) = [number of rows in the ORDER-LINE table for this district]*

for each district defined by (O_W_ID = OL_W_ID) and (O_D_ID = OL_D_ID).

4.2.5 Consistency Tests

Verify that the data base is initially consistent by verifying that it meets the consistency conditions defined in Clauses 3.3.2.1 to 3.3.2.4. Describe the steps used to do this in sufficient detail so that the steps are independently repeatable.

The consistency conditions defined in 4.2.1 through 4.2.4 were tested using a shell script to issue queries to the database. All queries showed that the data base was in a consistent state.

After executing transactions at full load for approximately sixty minutes the shell script was executed again. All queries show that the database was still in a consistent state.

4.3 Isolation Requirements

Operations of concurrent data base transactions must yield results which are indistinguishable from the results which would be obtained by forcing each transaction to be serially executed to completion in some order.

4.3.1 Isolation Test 1

This test demonstrates isolation for read-write conflicts of Order-Status and New-Order transactions.

1. An Order status transaction T0 was executed for a randomly selected customer, and the order returned was as recorded. Transaction T0 was committed.

2. A new-order transaction T1 was started for the same customer used in T0. T1 was stopped immediately prior to commit.

3. An order-status transaction T2 was started for the same customer used in T1. Transaction T2 completed and was committed without being blocked by T1. T2 returned the same order that T0 had returned.

4. T1 completed and was committed.

5. An order-status transaction T3 was started for the same customer used in T1. T3 returned the order inserted by T1.

This result demonstrates serialization of T2 before T1. It has equivalent validity to the outcome specified in the Standard which supposes T1 to be serialized before T2.

4.3.2 Isolation Test 2

This test demonstrates isolation for read-write conflicts of Order-Status and New-Order transactions when the New-Order transaction is rolled back.

The following steps were performed to satisfy the test of isolation for Order-Status and a rolled back New-Order transactions:

1. An Order status transaction T0 was executed for a randomly selected customer, and the order returned was recorded. Transaction T0 was committed.
2. A new-order transaction T1 with an invalid item was started for the same customer used in T0. Transaction T1 was stopped prior to rollback.
3. An order-status transaction T2 was started for the same customer used in T1. T2 completed and was committed without being blocked by T1. Transaction T2 returned the same order that T0 had returned.
4. T1 was rollback.
5. An order-status transaction T3 was started for the same customer used in T1. T3 returned the same order that T0 returned.

4.3.3 Isolation Test 3

This test demonstrates isolation for write-write conflicts of two New-Order transactions.

The following steps were performed to verify isolation of two New-Order transactions:

1. The D_NEXT_O_ID of a randomly selected district was retrieved.
2. A new-order transaction T1 was started for a randomly selected customer within the district used in step1. T1 was stopped immediately prior to commit.
3. Another new-order transaction was started for the same customer used in T1. Transaction T2 waited.
4. T1 completed. T2 completed and was committed.
5. The order number returned by T1 was the same as the D_NEXT_O_ID retrieved in step 1. The order number returned by T2 was one greater that the order number returned by T1.
6. The D_NEXT_O_ID of the same district was retrieved again. It had been incremented by two (it was one greater that the order number returned by T2).

4.3.4 Isolation Test 4

This test demonstrates isolation for write-write conflicts of two New-Order transactions when one transaction is rolled back.

The following steps were performed to verify the isolation of two New-Order transactions after one is rolled back:

1. The D_NEXT_O_ID of a randomly selected district was retrieved.
2. A new-order transaction T1 with an invalid item was started for a randomly selected customer with the district used in step1. T1 was stopped immediately prior to rollback.
3. Another new-order transaction was started for the same customer used in T1. T2 waited.
4. T1 was allowed to rollback. T2 completed and was committed.
5. The order number returned by T2 was the same as the D_NEXT_O_ID retrieved in step 1.
6. The D_NEXT_O_ID of the same district was retrieved again. It had been incremented by one (it was one greater that the order number returned by T2).

4.3.5 Isolation Test 5

This test demonstrates isolation for write-write conflicts of Payment and Delivery transactions.

The following steps were performed to successfully conduct this test:

1. A query was executed to find out the customer who would be updated by the next delivery transaction for a randomly selected warehouse and district.
2. The C_BALANCE of the customer found in step 1 is retrieved.
3. A delivery transaction T1 was started for the same warehouse used in step 1. T1 was stopped immediately prior to the commit of the database transaction corresponding to the district used in step 1.
4. A payment transaction T2 was started for the same customer found in step 1. T2 waited.
5. T1 was allowed to complete. T2 completed and was committed.
6. The C_BALANCE of the customer found in step 1 was retrieved again. The C_BALANCE reflected the results of both T1 and T2.

4.3.6 Isolation Test 6

This test demonstrates isolation for write-write conflicts of Payment and Delivery transactions when the Delivery transaction is rolled back.

The following steps were performed to successfully conduct this test:

1. A query was executed to find out the customer who would be updated by the next delivery transaction for a randomly selected warehouse and district.
2. The C_BALANCE of the customer found in step 1 is retrieved
3. A delivery transaction T1 was started for the same warehouse used in step 1. T1 was stopped immediately prior to the rollback of the database transaction corresponding to the district used in step 1.
4. A payment transaction T2 was started for the same customer found in step 1. Transaction T2 waited.
5. T1 was allowed to rollback. T2 completed and was committed.
6. The C_BALANCE of the customer found in step 1 was retrieved again. The C_BALANCE reflected the results of only Transaction T2.

4.3.7 Isolation Test 7

This test demonstrates repeatable reads for the New-Order transaction while an interactive transaction updates the price of an item.

The following steps were performed to successfully conduct this test:

1. The I_PRICE of two randomly selected items were retrieved.
2. A new-order transaction T2 with a group of items X and Y was started. T2 was stopped immediately, after retrieving the prices of all items. The prices of items X and Y retrieved matched those values retrieved in step 1.
3. A transaction T3 was started to increase the price of items X and Y by 10%.
4. T3 did not stall and no transaction was rolled back. T3 was committed.
5. T2 was resumed, and the prices of all items were retrieved again within T2. The prices of items X and Y matched those retrieved in step 1.
6. T2 was committed.
7. The prices of items X and Y were retrieved again. The values matched the values set by T3.

4.3.8 Isolation Test 8

This test demonstrates isolation for phantom protection between a Delivery and a New-Order transaction.

The following steps were performed to successfully conduct this test:

1. The NO_D_ID of all new order rows for a randomly selected warehouse and district was changed. The changes were committed.
2. A delivery transaction T1 was started for the selected customer.
3. T1 was stopped immediately after reading the new order table for the selected warehouse and district . No qualifying rows were found.
4. A new order transaction T2 was started for the same warehouse and district. T2 completed and was committed without being blocked by T1.
5. T1 was resumed and the new order table was read again. No qualifying row was found.
6. T1 completed and was committed.
7. The NO_D_ID of all new order rows for the selected warehouse and district was restored to the original value. The changes were committed.

4.3.9 Isolation Test 9

This test demonstrates isolation for phantom protection between an Order-Status and a New-Order transaction.

The following steps were performed to successfully conduct this test:

1. An order status transaction T1 was started for a randomly selected customer.
2. T1 was stopped immediately after reading the order table for the selected customer The most recent order for that customer was found.
3. A new order transaction T2 was started for the same customer. T2 completed and was committed without being blocked by T1.
4. T1 was resumed and the order table was read again to determine the most recent order for the same customer. The order found was the same as the one found in step 2.
5. T1 completed and was committed.

4.4 Durability Requirements

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

4.4.1 Permanent Unrecoverable Failure of any Single Durable Medium

Permanent irrecoverable failure of any single durable medium containing TPC-C data base tables or recovery log data.

Failure of Durable Medium containing recovery log data and Instantaneous Interruption and Memory Failure.

This test was conducted on a fully scaled database. The following steps were performed successfully.

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving SUM_1.
2. A test was started and allowed to run for twelve minutes.
3. One of the disks containing the transaction log data was powered off. Since the log was on a raid disk, Oracle 10g continued to process the transactions successfully.
4. The test continued for another 1 1/2 minutes.
5. The system was immediately shut down by switching the Emergency Power Off , thereby removing system

power.

6. The disk from step 3 was powered back on.
7. The system was powered back on and rebooted.
8. Step 1 is performed returning the value for SUM_2. It was verified that SUM_2 was equal to SUM_1 plus the completed New_Order transactions recorded by the RTE and that no entries existed for rolled-back transactions.
9. Consistency condition 3 was verified.

Failure of Durable Medium containing TPC-C data base tables.

The following steps were successfully performed to pass the Durability test of failure of a disk unit with data base tables:

1. The contents of a disk containing a TPCC table was backed up by copying it to another disk.
2. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving SUM_1.
3. A scaled-down test was started and allowed to run until steady state.
4. The disk containing the TPCC table was powered off.
5. The run was stopped.
6. The disk from step 4 was powered back on and was restored from the backup copy in step 1.
7. Oracle 10g was restarted and its transaction log was used to roll forward through the transactions that had completed since the run had started.
8. Step 2 was performed returning SUM_2. It was verified that SUM_2 was equal to SUM_1 plus the completed New_Order transactions recorded by the RTE and that no entries existed for rolled-back transactions.
9. Consistency condition 3 was verified.

Failure of Durable Fast Write Cache on FAStT700 Controller for data disks

These steps were performed on a fully scaled database to test the mirrored cache feature of the FAStT700

1. The current sum of all D_NEXT_O_ID values in the DIST table were obtained, giving sum1.
2. Workload was started and brought to steady state.
3. One of the two controller blade in the FAStT 700 was pulled out .
4. The test was allowed to complete.
5. A new sum of all D_NEXT_O_ID were sampled and, giving sum2.
6. Count of NewOrder transactions and rollbacks recorded by the RTE were obtained.
7. It was verified that no transactions were lost by the failure of one of the controllers by comparing sum2-sum1 with the count of NewOrder transactions recorded by the RTE, subtracting rollbacks.

Failure of Durable Fast Write Cache on SSA Adapter for Redo Logs

The following steps were successfully performed to pass the Durability test for failure of a durable medium that contains transient redo log transactions:

1. The SSA adapter for the Redo logs contains a cache that is powered by an onboard battery which will retain its contents if the adapter fails, or system power goes off. This test was performed in two parts:

   A) failure of the adapter/system power

   B) failure of the onboard battery

2. Test (A) was performed with the power-off test of the log above. After the system was powered off, the cache was removed from its current adapter and re-inserted into the system before rebooting.

3. Data on the cache was recovered successfully by meeting the requirements listed in the power-off test of the logs above.

4. Test (B). The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table, giving SUM_1.

5. Test (B) was conducted by inducing a battery failure during a test run. Once the system had reached steady state, the battery was failed using a toggle switch. The system recorded the failure, flushed out its vram contents and quit using the cache. Error notices were posted into the system error log.

6. The run continued without the cache.

7. Step 4 was performed returning SUM_2. It was verified that SUM_2 was equal to SUM_1 plus the completed New_Order transactions recorded by the RTE and that no entries existed for rolled-back transactions.

8. Consistency condition 3 was verified..

5. Clause 4: Scaling and Data Base Population Related Items

5.1 Cardinality of Tables

The cardinality (e.g., the number of rows) of each table, as it existed at the start of the benchmark run, must be disclosed.

Table 5-1 portrays the TPC Benchmark™ C defined tables and the number of rows for each table as they were built initially. Table 5-1 Initial Cardinality of Tables (eServer pSeries 690 Turbo Model 7040-681).

Table Name	Number of Rows
Warehouse	65,000
District	650,000
Customer	1,950,000,000
History	1,950,000,000
Order	1,950,000,000
New Order	585,000,000
Order Line	19,500,190,736
Stock	6,500,000,000
Item	100,000

5.2 Distribution of Tables and Logs

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

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

Table 5-2. IBM eServer pSeries 690 Turbo Model 7040-681 Data Distribution Benchmark Configuration

ADAPTER	HDISK	SIZE (GB)	LOGICAL VOLUMES
fcs0	hdisk11	67.64	lvaux1,lvdist2k1,lvdist1,lvhist1a,lvhist10,lvhist11a,lvhist12,lvfix_undo
fcs0	hdisk13,15,17,19	33.82	lvcust1,lvcust2,lvcust3,lvcust4,lvcust5,lvcust6,lvcust7,lvcust8,lvcust9,lvtemp1
fcs0	hdisk23,25,27,29,31	33.82	lvstk1,lvstk2,lvstk3,lvstk4,lvstk5,lvstk6,lvstk7,lvstk8,lvstk9,lvtemp2,lvstk640_n,lvstk641_n,lvstk642_n,lvstk643_n,lvstk644_n,lvstk645_n,lvstk646_n,lvstk647_n,lvstk648_n
fcs0	hdisk33,35,37,39,41	33.82	lvstk622,lvstk11,lvstk12,lvstk13,lvstk14,lvstk15,lvstk16,lvstk17,lvstk18,lvtemp3
fcs0	hdisk991	33.82	lvcust1,lvcust2,lvcust3,lvcust4,lvcust5,lvcust6,lvcust7,lvcust8,lvcust9,lvtemp1
fcs1	hdisk10	67.64	lvhist13,lvhist14,lvhist15,lvhist16,lvhist17,lvhist18,lvhist19
fcs1	hdisk12,14,16,18,20	33.82	lvcust10,lvcust11,lvcust12,lvcust13,lvcust14,lvcust15,lvcust16,lvcust17,lvcust18,lvtemp4
fcs1	hdisk24,26,28,30,1185	33.82	lvstk19,lvstk667,lvstk21,lvstk22,lvstk23,lvstk24,lvstk25,lvstk26,lvstk27,lvtemp5
fcs1	hdisk32	33.82	lvstk28,lvstk29,lvstk30,lvstk31,lvstk32,lvstk33,lvstk34,lvstk35,lvstk36,lvtemp6,lvaux1_s,lvdist2k1_s,lvdist1_s,lvhist10_s,lvhist12_s
fcs1	hdisk36,38,40,1186	33.82	lvstk28,lvstk29,lvstk30,lvstk31,lvstk32,lvstk33,lvstk34,lvstk35,lvstk36,lvtemp6
fcs2	hdisk43	67.64	lvi1cu3,lvi2cu7,lvi2cu8,lvi2cu9,lvistk3
fcs2	hdisk45,47,49,51,53	33.82	lvcust109,lvcust321a,lvcust111,lvcust112,lvcust113,lvcust114,lvcust115,lvcust116,lvcust117,lvtemp37

fcs2	hdisk55,57,59,61,63	33.82	lvstk217,lvstk218,lvstk219,lvstk220,lvstk221,lvstk222,lvstk223,lvstk224,lvstk225,lvtemp38
fcs2	hdisk65,67,69,71,73	33.82	lvstk226,lvstk227,lvstk228,lvstk229,lvstk230,lvstk231,lvstk232,lvstk233,lvstk234,lvtemp39
fcs3	hdisk42	67.64	lvi1cu8,lvi2cu22,lvi2cu23,lvistk8,lvistk17,lvi2cu24
fcs3	hdisk44,46,48,50,52	33.82	lvcust154,lvcust155,lvcust156,lvcust157,lvcust158,lvcust159,lvcust160,lvcust161,lvcust162,lvtemp52
fcs3	hdisk54,56,58,60,62	33.82	lvstk307,lvstk308,lvstk309,lvstk310,lvstk311,lvstk312,lvstk313,lvstk314,lvstk315,lvtemp53
fcs3	hdisk64,66,68,70,72	33.82	lvstk316,lvstk317,lvstk318,lvstk319,lvstk320,lvstk322,lvstk323,lvstk324,lvtemp54
fcs4	hdisk74	67.64	lvi2or1,lvi2or2,lvi2or3,lvi2or4
fcs4	hdisk76,78,80,82,84	33.82	lvcust163,lvcust164,lvcust165,lvcust166,lvcust167,lvcust168,lvcust169,lvcust170,lvcust171,lvtemp55
fcs4	hdisk86,88,90,92,94	33.82	lvstk325,lvstk326,lvstk327,lvstk328,lvstk329,lvstk330,lvstk331,lvstk332,lvstk333,lvtemp56
fcs4	hdisk96,98,100,102,104	33.82	lvstk334,lvstk335,lvstk336,lvstk337,lvstk338,lvstk339,lvstk340,lvstk341,lvstk342,lvtemp57
fcs5	hdisk106	67.64	lvi2or45,lvi2or46,lvi2or47,lvi2or48
fcs5	hdisk108,110,112,114,116	33.82	lvcust262,lvcust263,lvcust264,lvcust265,lvcust266,lvcust267,lvcust268,lvcust269,lvcust270,lvtemp88
fcs5	hdisk118,120,122,124,126	33.82	lvstk523,lvstk524,lvstk525,lvstk526,lvstk527,lvstk528,lvstk529,lvstk530,lvstk531,lvtemp89
fcs5	hdisk128,130,132,134,136	33.82	lvstk532,lvstk533,lvstk534,lvstk535,lvstk536,lvstk537,lvstk538,lvstk539,lvstk540,lvtemp90
fcs6	hdisk107	67.64	lvi2or89,lvi2or90,lvi2or91,lvi2or92
fcs6	hdisk109,111,113,115,117	33.82	lvcust361,lvcust362,lvcust363,lvcust364,lvcust365,lvcust366,lvcust367,lvcust368,lvcust369,lvtemp121
fcs6	hdisk119,121,123,125,127	33.82	lvstk721,lvstk722,lvstk723,lvstk724,lvstk725,lvstk726,lvstk727,lvstk728,lvstk729,lvtemp122
fcs6	hdisk129,131,133,135,137	33.82	lvstk730,lvstk731,lvstk732,lvstk733,lvstk734,lvstk735,lvstk736,lvstk737,lvstk738,lvtemp123
fcs7	hdisk140,142,144,146,148	33.82	lvcust460,lvcust461,lvcust462,lvcust463,lvcust464,lvcust465,lvhist11,lvnord13,lvnord14,lvnord15,lvcust360,lvcust359,lvcust358,lvcust357
fcs7	hdisk150,152,154,156,158	33.82	lvstk919,lvstk920,lvstk921,lvstk922,lvstk923,lvstk924,lvstk925,lvstk926,lvstk927
fcs7	hdisk160,162,164,166,168	33.82	lvstk928,lvstk929,lvstk930,lvstk931,lvstk932,lvstk933,lvstk934,lvstk935,lvstk936
fcs8	hdisk141,143,145,147,149	33.82	lvnord1,lvnord10,lvnord11,lvnord12,lvordl204,lvordl205,lvordl206,lvordl207,lvordl208,lvordl209
fcs8	hdisk151,153	33.82	lvi1cu5,lvi2cu13,lvi2cu14,lvi2cu15,lvistk5,lvistk14
fcs9	hdisk75	67.64	lvi1cu9,lvistk9,lvistk18,lvi2cu25,lvi2cu26,lvi2cu27
fcs9	hdisk77,79,81,83,85	33.82	lvordl213,lvordl214,lvordl215,lvordl216,lvordl201,lvnord5,lvnord6,lvnord2,lvnord3,lvnord4
fcs9	hdisk87,89	33.82	lvi1cu3,lvi2cu7,lvi2cu8,lvi2cu9,lvistk3
fcs9	hdisk101,103	33.82	lvistk12,lvi1cu4,lvi2cu10,lvi2cu11,lvi2cu12,lvistk4,lvistk13
fcs10	hdisk171	67.64	lvhist2,lvhist20,lvhist21,lvhist22,lvhist23,lvhist24,lvhist3
fcs10	hdisk173,175,177,179,181	33.82	lvcust19,lvcust20,lvcust21,lvcust22,lvcust23,lvcust24,lvcust25,lvcust26,lvcust27,lvtemp7
fcs10	hdisk183,185,187,189,191	33.82	lvstk37,lvstk38,lvstk39,lvstk40,lvstk41,lvstk42,lvstk43,lvstk44,lvstk45,lvtemp8
fcs10	hdisk193,195,197,199,201	33.82	lvstk46,lvstk47,lvstk48,lvstk49,lvstk50,lvstk51,lvstk52,lvstk53,lvstk54,lvtemp9
fcs11	hdisk170	67.64	lvidist1,lvidist2k1,lvhist4,lvhist6,lvhist7,lvhist9,lvhist5,lvhist8
fcs11	hdisk172,174,176,178,180	33.82	lvcust28,lvcust29,lvcust30,lvcust31,lvcust32,lvcust33,lvcust34,lvcust35,lvcust36,lvtemp10
fcs11	hdisk182,184,186,188,190	33.82	lvstk55,lvstk56,lvstk57,lvstk58,lvstk59,lvstk60,lvstk61,lvstk62,lvstk63
fcs11	hdisk192,194,196,198,200	33.82	lvstk64,lvstk65,lvstk66,lvstk67,lvstk68,lvstk69,lvstk70,lvstk71,lvstk72
fcs12	hdisk202	67.64	lviitem1,lvitem1,lviware1,lvnord1a,lvnord10a,lvnord11a,lvnord12a,lvnord13a,lvnord14a,lvnord15a,lvsblog1,lvsblog2,lv2kware1,lv2kiware1

fcs12	hdisk204,206,208,210,212	33.82	lvcust37,lvcust38,lvcust39,lvcust40,lvcust41,lvcust42,lvcust43,lvcust44,lvcust45,lvtemp13
fcs12	hdisk214,216,218,220,222	33.82	lvstk73,lvstk74,lvstk75,lvstk76,lvstk77,lvstk78,lvstk79,lvstk80,lvstk81,lvtemp14
fcs12	hdisk224,226,228,230,232	33.82	lvstk82,lvstk83,lvstk84,lvstk85,lvstk86,lvstk87,lvstk88,lvstk89,lvstk90,lvtemp15
fcs13	hdisk205,207,209,211,213	33.82	lvcust46,lvcust47,lvcust48,lvcust49,lvcust50,lvcust51,lvcust52,lvcust53,lvcust54
fcs13	hdisk215,217,219,221,223	33.82	lvstk91,lvstk92,lvstk93,lvstk94,lvstk95,lvstk96,lvstk97,lvstk98,lvstk99,lvtemp17
fcs13	hdisk225,227,229,231,233	33.82	lvstk100,lvstk101,lvstk102,lvstk103,lvstk104,lvstk1007,lvstk106,lvstk107,lvstk108,lvtemp18
fcs14	hdisk237,239,241,243,245	33.82	lvcust55,lvcust56,lvcust57,lvcust58,lvcust59,lvcust60,lvcust61,lvcust62,lvcust63
fcs14	hdisk247,249,251,253,255	33.82	lvstk109,lvstk110,lvstk111,lvstk112,lvstk113,lvstk114,lvstk115,lvstk116,lvstk117
fcs14	hdisk257,259,261,263,265	33.82	lvstk118,lvstk119,lvstk120,lvstk121,lvstk122,lvstk123,lvstk124,lvstk125,lvstk126,lvtemp21
fcs15	hdisk268,270,272,274,276	33.82	lvcust64,lvcust65,lvcust66,lvcust67,lvcust68,lvcust69,lvcust70,lvcust71,lvcust72,lvtemp22
fcs15	hdisk278,280,282,284,286	33.82	lvstk127,lvstk128,lvstk129,lvstk130,lvstk131,lvstk132,lvstk133,lvstk134,lvstk135,lvtemp23
fcs15	hdisk288,294,368,370,372	33.82	lvstk136,lvstk137,lvstk138,lvstk139,lvstk140,lvstk141,lvstk142,lvstk143,lvstk144,lvtemp24
fcs16	hdisk269,271,273,275,277	33.82	lvcust73,lvcust74,lvcust75,lvcust76,lvcust77,lvcust78,lvcust79,lvcust80,lvcust81
fcs16	hdisk279,281,283,285,287	33.82	lvstk145,lvstk146,lvstk147,lvstk148,lvstk149,lvstk150,lvstk151,lvstk152,lvstk153,lvtemp26
fcs16	hdisk289,367,369,371,373	33.82	lvstk154,lvstk155,lvstk156,lvstk157,lvstk158,lvstk159,lvstk160,lvstk161,lvstk162,lvtemp27
fcs17	hdisk374	67.64	lvsp1,lvsystem1,lvsystem2,lvsp2,lvsp3
fcs17	hdisk376,378,380,382,384	33.82	lvcust82,lvcust83,lvcust84,lvcust85,lvcust86,lvcust87,lvcust88,lvcust89,lvcust90,lvtemp28
fcs17	hdisk386,388,390,392,401	33.82	lvstk163,lvstk164,lvstk165,lvstk166,lvstk167,lvstk168,lvstk169,lvstk170,lvstk171,lvtemp29
fcs17	hdisk403,405,407,409,411	33.82	lvstk172,lvstk173,lvstk174,lvstk175,lvstk176,lvstk177,lvstk178,lvstk179,lvstk180,lvtemp30
fcs18	hdisk375	67.64	lvi1cu1,lvi2cu2,lvi2cu3,lvistk1,lvistk10,lvi2cu1
fcs18	hdisk377,379,381,383,385	33.82	lvcust91,lvcust92,lvcust93,lvcust94,lvcust95,lvcust96,lvcust97,lvcust98,lvcust99,lvtemp31
fcs18	hdisk387,389,391,400,402	33.82	lvstk181,lvstk182,lvstk183,lvstk184,lvstk185,lvstk186,lvstk187,lvstk188,lvstk189,lvtemp32
fcs18	hdisk404,406,408,410,412	33.82	lvstk190,lvstk191,lvstk192,lvstk193,lvstk194,lvstk195,lvstk196,lvstk197,lvstk198,lvtemp33
fcs19	hdisk234	67.64	lvi1cu2,lvi2cu4,lvi2cu5,lvi2cu6,lvistk2,lvistk11
fcs19	hdisk236,238,240,242,244	33.82	lvcust100,lvcust101,lvcust102,lvcust103,lvcust104,lvcust105,lvcust106,lvcust107,lvcust108,lvtemp34
fcs19	hdisk246,248,250,252,254	33.82	lvstk199,lvstk200,lvstk201,lvstk202,lvstk203,lvstk204,lvstk205,lvstk206,lvstk207,lvtemp35
fcs19	hdisk256,258,260,262,264	33.82	lvstk208,lvstk209,lvstk210,lvstk211,lvstk212,lvstk213,lvstk214,lvstk215,lvstk216,lvtemp36
fcs20	hdisk413	67.64	lvistk12,lvi1cu4,lvi2cu10,lvi2cu11,lvi2cu12,lvistk4,lvistk13
fcs20	hdisk415,417,419,421,423	33.82	lvcust118,lvcust119,lvcust120,lvcust121,lvcust122,lvcust123,lvcust124,lvcust125,lvcust126,lvtemp40
fcs20	hdisk425,427,429,431,433	33.82	lvstk235,lvstk236,lvstk237,lvstk238,lvstk239,lvstk240,lvstk241,lvstk242,lvstk243,lvtemp41
fcs20	hdisk435,437,439,441,443	33.82	lvstk244,lvstk245,lvstk246,lvstk247,lvstk248,lvstk249,lvstk250,lvstk251,lvstk252,lvtemp42
fcs21	hdisk414	67.64	lvi1cu5,lvi2cu13,lvi2cu14,lvi2cu15,lvistk5,lvistk14
fcs21	hdisk416,418,420,422,424	33.82	lvcust127,lvcust128,lvcust129,lvcust130,lvcust131,lvcust132,lvcust133,lvcust134,lvcust135,lvtemp43
fcs21	hdisk426,428,430,432,434	33.82	lvstk253,lvstk254,lvstk255,lvstk256,lvstk257,lvstk258,lvstk259,lvstk260,lvstk261,lvtemp44
fcs21	hdisk436,438,440,442,444	33.82	lvstk262,lvstk263,lvstk264,lvstk265,lvstk266,lvstk267,lvstk268,lvstk631,lvstk270,lvtemp45
fcs22	hdisk445	67.64	lvi1cu6,lvi2cu16,lvi2cu17,lvi2cu18,lvistk6,lvistk15

fcs22	hdisk447,449,451,453,455	33.82	lvcust136,lvcust137,lvcust138,lvcust139,lvcust140,lvcust141,lvcust142,lvcust143,lvcust144,lvtemp46
fcs22	hdisk457,459,461,463,465	33.82	lvstk271,lvstk272,lvstk273,lvstk274,lvstk275,lvstk276,lvstk277,lvstk278,lvstk279,lvtemp47
fcs22	hdisk467,469,471,473,475	33.82	lvstk280,lvstk281,lvstk282,lvstk283,lvstk284,lvstk285,lvstk286,lvstk287,lvstk288,lvtemp48
fcs23	hdisk446	67.64	lvi1cu7,lvi2cu19,lvi2cu20,lvi2cu21,lvistk7,lvistk16
fcs23	hdisk448,450,452,454,456	33.82	lvcust145,lvcust146,lvcust147,lvcust148,lvcust149,lvcust150,lvcust151,lvcust152,lvcust153,lvtemp49
fcs23	hdisk458,460,462,464,466	33.82	lvstk289,lvstk290,lvstk291,lvstk292,lvstk293,lvstk294,lvstk295,lvstk296,lvstk297,lvtemp50
fcs23	hdisk468,470,472,474,476	33.82	lvstk298,lvstk299,lvstk300,lvstk301,lvstk302,lvstk303,lvstk304,lvstk305,lvstk306
fcs30	hdisk519,520	33.82	lvi1cu6,lvi2cu16,lvi2cu17,lvi2cu18,lvistk6,lvistk15
fcs30	hdisk537,538	33.82	lvi1cu7,lvi2cu19,lvi2cu20,lvi2cu21,lvistk7,lvistk16
fcs30	hdisk541,542	33.82	lvi1cu8,lvi2cu22,lvi2cu23,lvistk8,lvistk17,lvi2cu24
fcs31	hdisk512	67.64	lvi1cu2,lvi2cu4,lvi2cu5,lvi2cu6,lvistk2,lvistk11
fcs40	hdisk395	33.82	lvstk343,lvstk344,lvstk345,lvstk346,lvstk347,lvstk348,lvstk349,lvstk350,lvstk351
fcs40	hdisk543	67.64	lvi2or5,lvi2or6,lvi2or7,lvi2or8
fcs40	hdisk545,547,549,551,553	33.82	lvcust172,lvcust173,lvcust174,lvcust175,lvcust176,lvcust177,lvcust178,lvcust179,lvcust180,lvtemp58
fcs40	hdisk557,559,561,563	33.82	lvstk343,lvstk344,lvstk345,lvstk346,lvstk347,lvstk348,lvstk349,lvstk350,lvstk351
fcs40	hdisk565,567,569,571,573	33.82	lvstk352,lvstk353,lvstk354,lvstk355,lvstk356,lvstk357,lvstk358,lvstk359,lvstk360,lvtemp60
fcs41	hdisk394	33.82	lvstk361,lvstk362,lvstk363,lvstk364,lvstk365,lvstk366,lvstk367,lvstk368,lvstk369
fcs41	hdisk544	67.64	lvi2or9,lvi2or10,lvi2or11,lvi2or12
fcs41	hdisk546,548,550,552,554	33.82	lvcust181,lvcust182,lvcust183,lvcust184,lvcust185,lvcust186,lvcust187,lvcust188,lvcust189,lvtemp61
fcs41	hdisk556,558,560,564	33.82	lvstk361,lvstk362,lvstk363,lvstk364,lvstk365,lvstk366,lvstk367,lvstk368,lvstk369
fcs41	hdisk566,568,570,572,574	33.82	lvstk370,lvstk371,lvstk372,lvstk373,lvstk374,lvstk375,lvstk376,lvstk377,lvstk378,lvtemp63
fcs42	hdisk576	67.64	lvi2or13,lvi2or14,lvi2or15,lvi2or16
fcs42	hdisk578,580,582,584,586	33.82	lvcust190,lvcust191,lvcust192,lvcust193,lvcust194,lvcust195,lvcust196,lvcust197,lvcust198,lvtemp64
fcs42	hdisk588,590,592,594,596	33.82	lvstk379,lvstk380,lvstk381,lvstk382,lvstk383,lvstk384,lvstk385,lvstk386,lvstk387
fcs42	hdisk598,600,602,604,606	33.82	lvstk388,lvstk389,lvstk390,lvstk391,lvstk392,lvstk393,lvstk394,lvstk395,lvstk396,lvtemp66
fcs43	hdisk575	67.64	lvi2or17,lvi2or18,lvi2or19,lvi2or20
fcs43	hdisk577,579,581,583,585	33.82	lvcust199,lvcust200,lvcust201,lvcust202,lvcust203,lvcust204,lvcust205,lvcust206,lvcust207,lvcust110,lvtemp67
fcs43	hdisk587,589,591,593,595	33.82	lvstk397,lvstk398,lvstk399,lvstk400,lvstk401,lvstk402,lvstk403,lvstk404,lvstk405,lvtemp68
fcs43	hdisk597,599,601,603,605	33.82	lvstk406,lvstk407,lvstk408,lvstk409,lvstk410,lvstk411,lvstk412,lvstk413,lvstk414,lvtemp69
fcs44	hdisk607	67.64	lvi2or21,lvi2or22,lvi2or23,lvi2or24
fcs44	hdisk609,611,613,615,617	33.82	lvcust208,lvcust209,lvcust210,lvcust211,lvcust212,lvcust213,lvcust214,lvcust215,lvcust216,lvtemp70
fcs44	hdisk619,621,623,625,627	33.82	lvstk415,lvstk416,lvstk417,lvstk418,lvstk419,lvstk489,lvstk421,lvstk422,lvstk423,lvtemp71
fcs44	hdisk629,631,633,635,637	33.82	lvstk424,lvstk425,lvstk426,lvstk427,lvstk428,lvstk429,lvstk430,lvstk431,lvstk432,lvtemp72
fcs45	hdisk640	67.64	lvi2or25,lvi2or26,lvi2or27,lvi2or28
fcs45	hdisk642,644,646,648,650	33.82	lvcust217,lvcust218,lvcust219,lvcust220,lvcust221,lvcust222,lvcust223,lvcust224,lvcust225,lvtemp73
fcs45	hdisk652,654,656,658,660	33.82	lvstk433,lvstk434,lvstk435,lvstk436,lvstk437,lvstk438,lvstk865,lvstk440,lvstk441,lvtemp74
fcs45	hdisk662,664,666,668,670	33.82	lvstk442,lvstk443,lvstk444,lvstk445,lvstk446,lvstk447,lvstk448,lvstk449,lvstk450,lvtemp75
fcs46	hdisk639	67.64	lvi2or29,lvi2or30,lvi2or31,lvi2or32
fcs46	hdisk641,643,645,647,649	33.82	lvcust226,lvcust227,lvcust228,lvcust229,lvcust230,lvcust231,lvcust232,lvcust233,lvcust234,lvtemp76

fcs46	hdisk651,653,655,657,659	33.82	lvstk451,lvstk452,lvstk453,lvstk454,lvstk455,lvstk456,lvstk457,lvstk458,lvstk459,lvtemp77
fcs46	hdisk661,663,665,667,669	33.82	lvstk460,lvstk461,lvstk462,lvstk463,lvstk464,lvstk465,lvstk466,lvstk467,lvstk468,lvtemp78
fcs47	hdisk672	67.64	lvi2or33,lvi2or34,lvi2or35,lvi2or36
fcs47	hdisk674,676,678,680,682	33.82	lvcust235,lvcust236,lvcust237,lvcust238,lvcust239,lvcust240,lvcust241,lvcust242,lvcust243,lvtemp79
fcs47	hdisk684,686,688,690,692	33.82	lvstk469,lvstk470,lvstk471,lvstk472,lvstk473,lvstk474,lvstk475,lvstk476,lvstk477,lvtemp80
fcs47	hdisk694,696,698,700,702	33.82	lvstk478,lvstk479,lvstk480,lvstk481,lvstk482,lvstk483,lvstk484,lvstk485,lvstk486,lvtemp81
fcs48	hdisk671,	67.64	lvi2or37,lvi2or38,lvi2or39,lvi2or40
fcs48	hdisk681,673,675,677,679	33.82	lvcust244,lvcust245,lvcust246,lvcust247,lvcust248,lvcust249,lvcust250,lvcust251,lvcust252,lvtemp82
fcs48	hdisk683,685,687,689,691	33.82	lvstk487,lvstk488,lvstk420,lvstk490,lvstk491,lvstk492,lvstk493,lvstk494,lvstk495,lvtemp83
fcs48	hdisk693,695,697,699,701	33.82	lvstk496,lvstk497,lvstk498,lvstk499,lvstk500,lvstk501,lvstk502,lvstk503,lvstk504,lvtemp84
fcs49	hdisk608	67.64	lvi2or41,lvi2or42,lvi2or43,lvi2or44
fcs49	hdisk610,612,614,616,618	33.82	lvcust253,lvcust254,lvcust255,lvcust256,lvcust257,lvcust258,lvcust259,lvcust260,lvcust261,lvtemp85
fcs49	hdisk620,622,624,626,628	33.82	lvstk505,lvstk506,lvstk507,lvstk508,lvstk509,lvstk510,lvstk511,lvstk512,lvstk513,lvtemp86
fcs49	hdisk630,632,634,636,638	33.82	lvstk514,lvstk515,lvstk516,lvstk517,lvstk518,lvstk519,lvstk520,lvstk521,lvstk522,lvtemp87
fcs50	hdisk703	67.64	lvi2or49,lvi2or50,lvi2or51,lvi2or52
fcs50	hdisk705,707,709,711,713	33.82	lvcust271,lvcust272,lvcust273,lvcust274,lvcust275,lvcust276,lvcust277,lvcust278,lvcust279,lvtemp91
fcs50	hdisk715,717,719,721,723	33.82	lvstk541,lvstk542,lvstk543,lvstk544,lvstk545,lvstk546,lvstk547,lvstk548,lvstk549,lvtemp92
fcs50	hdisk725,727,729,731,733	33.82	lvstk550,lvstk551,lvstk552,lvstk553,lvstk554,lvstk555,lvstk556,lvstk557,lvstk558,lvtemp93
fcs51	hdisk704	67.64	lvi2or53,lvi2or54,lvi2or55,lvi2or56
fcs51	hdisk706,708,710,712,714	33.82	lvcust280,lvcust281,lvcust282,lvcust283,lvcust284,lvcust285,lvcust286,lvcust287,lvcust288,lvtemp94
fcs51	hdisk716,718,720,722,724	33.82	lvstk956,lvstk560,lvstk561,lvstk562,lvstk563,lvstk564,lvstk565,lvstk566,lvstk567,lvtemp95
fcs51	hdisk726,728,730,732,734	33.82	lvstk568,lvstk569,lvstk570,lvstk571,lvstk572,lvstk573,lvstk574,lvstk575,lvstk576,lvtemp96
fcs52	hdisk736	67.64	lvi2or57,lvi2or58,lvi2or59,lvi2or60
fcs52	hdisk738,740,742,744,746	33.82	lvcust289,lvcust290,lvcust291,lvcust292,lvcust293,lvcust294,lvcust295,lvcust296,lvcust297,lvtemp97
fcs52	hdisk748,750,752,754,756	33.82	lvstk577,lvstk578,lvstk579,lvstk580,lvstk581,lvstk582,lvstk584,lvstk585,lvtemp98
fcs52	hdisk758,760,762,764,766	33.82	lvstk586,lvstk587,lvstk588,lvstk589,lvstk590,lvstk591,lvstk592,lvstk593,lvstk594,lvtemp99
fcs53	hdisk735	67.64	lvi2or61,lvi2or62,lvi2or63,lvi2or64
fcs53	hdisk737,739,741,743,745	33.82	lvcust298,lvcust299,lvcust300,lvcust301,lvcust302,lvcust303,lvcust304,lvcust305,lvcust306,lvtemp100
fcs53	hdisk747,749,751,753,755	33.82	lvstk595,lvstk596,lvstk597,lvstk598,lvstk599,lvstk600,lvstk601,lvstk602,lvstk603,lvtemp101,lvstk937
fcs53	hdisk757,759,761,763,765	33.82	lvstk604,lvstk605,lvstk606,lvstk607,lvstk608,lvstk609,lvstk610,lvstk611,lvstk612,lvtemp102
fcs54	hdisk768	67.64	lvi2or65,lvi2or66,lvi2or67,lvi2or68
fcs54	hdisk770,772,774,776,778	33.82	lvcust307,lvcust308,lvcust309,lvcust310,lvcust311,lvcust312,lvcust313,lvcust314,lvcust315,lvtemp103
fcs54	hdisk780,782,784,786,788	33.82	lvstk613,lvstk614,lvstk615,lvstk616,lvstk617,lvstk618,lvstk619,lvstk620,lvstk621,lvtemp104
fcs54	hdisk790,792,794,796,798	33.82	lvstk10,lvstk623,lvstk624,lvstk625,lvstk626,lvstk627,lvstk628,lvstk629,lvstk630,lvtemp105
fcs55	hdisk799	67.64	lvi2or69,lvi2or70,lvi2or71,lvi2or72
fcs55	hdisk801,803,805,807,809	33.82	lvcust316,lvcust317,lvcust318,lvcust319,lvcust320,lvcust321,lvcust322,lvcust323,lvcust324,lvtemp106
fcs55	hdisk811,813,815,817,819	33.82	lvstk269,lvstk632,lvstk633,lvstk634,lvstk635,lvstk636,lvstk637,lvstk638,lvstk639,lvtemp107

fcs55	hdisk821,823,825,827,829	33.82	lvstk640,lvstk641,lvstk642,lvstk643,lvstk644,lvstk645,lvstk646,lvstk647,lvstk648,lvtemp108
fcs56	hdisk800	67.64	lvi2or73,lvi2or74,lvi2or75,lvi2or76
fcs56	hdisk802,804,806,808,810	33.82	lvcust325,lvcust326,lvcust327,lvcust328,lvcust329,lvcust330,lvcust331,lvcust332,lvcust333,lvtemp109
fcs56	hdisk812,814,816,818,820	33.82	lvstk649,lvstk650,lvstk651,lvstk652,lvstk653,lvstk654,lvstk655,lvstk656,lvstk657,lvtemp110
fcs56	hdisk822,824,826,828,830	33.82	lvstk658,lvstk659,lvstk660,lvstk661,lvstk662,lvstk663,lvstk664,lvstk665,lvstk666,lvtemp111
fcs57	hdisk831	67.64	lvi2or77,lvi2or78,lvi2or79,lvi2or80
fcs57	hdisk833,835,837,839,841	33.82	lvcust334,lvcust335,lvcust336,lvcust337,lvcust338,lvcust339,lvcust340,lvcust341,lvcust342,lvtemp112
fcs57	hdisk843,845,847,849,851	33.82	lvstk20,lvstk668,lvstk669,lvstk670,lvstk671,lvstk672,lvstk673,lvstk674,lvstk675,lvtemp113
fcs57	hdisk853,855,857,859,861	33.82	lvstk676,lvstk677,lvstk678,lvstk679,lvstk680,lvstk681,lvstk682,lvstk683,lvstk684,lvtemp114
fcs58	hdisk832	67.64	lvi2or81,lvi2or82,lvi2or83,lvi2or84
fcs58	hdisk834,836,838,840,842	33.82	lvcust343,lvcust344,lvcust345,lvcust346,lvcust347,lvcust348,lvcust349,lvcust350,lvcust351,lvtemp115
fcs58	hdisk844,846,848,850,852	33.82	lvstk685,lvstk686,lvstk687,lvstk688,lvstk689,lvstk690,lvstk691,lvstk692,lvstk693,lvtemp116
fcs58	hdisk854,856,858,860,862	33.82	lvstk694,lvstk695,lvstk696,lvstk697,lvstk698,lvstk699,lvstk700,lvstk701,lvstk702,lvtemp117
fcs59	hdisk767	67.64	lvi2or85,lvi2or86,lvi2or87,lvi2or88
fcs59	hdisk769,771,773,775,777	33.82	lvcust352,lvcust353,lvcust354,lvcust355,lvcust356,lvcust357a,lvcust358a,lvcust359a,lvcust360a,lvtemp118
fcs59	hdisk779,781,783,785,787	33.82	lvstk703,lvstk704,lvstk705,lvstk706,lvstk707,lvstk708,lvstk709,lvstk710,lvstk711,lvstk321,lvtemp119
fcs59	hdisk789,791,793,795,797	33.82	lvstk712,lvstk713,lvstk714,lvstk715,lvstk716,lvstk717,lvstk718,lvstk719,lvstk720,lvtemp120
fcs60	hdisk864	67.64	lvi2or93,lvi2or94,lvi2or95,lvi2or96
fcs60	hdisk866,868,870,872,874	33.82	lvcust370,lvcust371,lvcust372,lvcust373,lvcust374,lvcust375,lvcust376,lvcust377,lvcust378
fcs60	hdisk876,878,880,882,884	33.82	lvstk739,lvstk740,lvstk741,lvstk742,lvstk743,lvstk744,lvstk745,lvstk746,lvstk747,lvtemp125
fcs60	hdisk886,888,890,892,894	33.82	lvstk748,lvstk749,lvstk750,lvstk751,lvstk752,lvstk753,lvstk754,lvstk755,lvstk756,lvtemp126
fcs61	hdisk895	67.64	lvi2or97,lvi2or98,lvi2or99,lvi2or100
fcs61	hdisk897,899,901,903,905	33.82	lvcust379,lvcust380,lvcust381,lvcust382,lvcust383,lvcust384,lvcust385,lvcust386,lvcust387,lvtemp127
fcs61	hdisk907,909,911,913,915	33.82	lvstk757,lvstk758,lvstk759,lvstk760,lvstk761,lvstk762,lvstk763,lvstk764,lvstk765
fcs61	hdisk917,919,921,923,925	33.82	lvstk766,lvstk767,lvstk768,lvstk769,lvstk770,lvstk771,lvstk772,lvstk773,lvstk774,lvtemp129
fcs62	hdisk928	67.64	lvi2or101,lvi2or102,lvi2or103,lvi2or104
fcs62	hdisk930,932,934,936,938	33.82	lvcust388,lvcust389,lvcust390,lvcust391,lvcust392,lvcust393,lvcust394,lvcust395,lvcust396,lvtemp130
fcs62	hdisk940,942,944,946,948	33.82	lvstk775,lvstk776,lvstk777,lvstk778,lvstk779,lvstk780,lvstk781,lvstk782,lvstk783,lvtemp131
fcs62	hdisk950,952,954,956,958	33.82	lvstk784,lvstk785,lvstk786,lvstk787,lvstk788,lvstk789,lvstk790,lvstk791,lvstk792,lvtemp132
fcs63	hdisk927	67.64	lvi2or105,lvi2or106,lvi2or107,lvi2or108
fcs63	hdisk929,931,933,935,937	33.82	lvcust397,lvcust398,lvcust399,lvcust400,lvcust401,lvcust402,lvcust403,lvcust404,lvcust405,lvtemp133
fcs63	hdisk939,941,943,945,947	33.82	lvstk793,lvstk794,lvstk795,lvstk796,lvstk797,lvstk798,lvstk799,lvstk800,lvstk801,lvtemp134
fcs63	hdisk949,951,953,955,957	33.82	lvstk802,lvstk803,lvstk804,lvstk805,lvstk806,lvstk807,lvstk808,lvstk809,lvstk810,lvtemp135,lvware1
fcs64	hdisk960	67.64	lvi2or109,lvi2or110,lvi2or111,lvi2or112
fcs64	hdisk962,964,966,968,970	33.82	lvcust406,lvcust407,lvcust408,lvcust409,lvcust410,lvcust411,lvcust412,lvcust413,lvcust414
fcs64	hdisk972,974,976,978,980	33.82	lvstk811,lvstk812,lvstk813,lvstk814,lvstk815,lvstk816,lvstk817,lvstk818,lvstk819
fcs64	hdisk982,984,986,988,990	33.82	lvstk820,lvstk821,lvstk822,lvstk823,lvstk824,lvstk825,lvstk826,lvstk827,lvstk828,lvtemp138
fcs65	hdisk992	67.64	lvi2or113,lvi2or114,lvi2or115,lvi2or116

fcs65	hdisk994,996,998,1000,1002	33.82	lvcust415,lvcust416,lvcust417,lvcust418,lvcust419,lvcust420,lvcust421,lvcust422,lvcust423,lvtemp139
fcs65	hdisk1004,1006,1008,1010,1012	33.82	lvstk829,lvstk830,lvstk831,lvstk832,lvstk833,lvstk834,lvstk835,lvstk836,lvstk837,lvtemp140
fcs65	hdisk1014,1016,1018,1020,1022	33.82	lvstk838,lvstk839,lvstk840,lvstk841,lvstk842,lvstk843,lvstk844,lvstk845,lvstk846
fcs66	hdisk993	67.64	lvi2or117,lvi2or118,lvi2or119,lvi2or120
fcs66	hdisk995,997,999,1001,1003	33.82	lvcust424,lvcust425,lvcust426,lvcust427,lvcust428,lvcust429,lvcust430,lvcust431,lvcust432
fcs66	hdisk1005,1007,1009,1011,1013	33.82	lvstk847,lvstk848,lvstk849,lvstk850,lvstk851,lvstk852,lvstk853,lvstk854,lvstk855
fcs66	hdisk1015,1017,1019,1021,1183	33.82	lvstk856,lvstk857,lvstk858,lvstk859,lvstk860,lvstk861,lvstk862,lvstk863,lvstk864
fcs67	hdisk1025,1027,1029,1031,1033	33.82	lvcust433,lvcust434,lvcust435,lvcust436,lvcust437,lvcust438,lvcust439,lvcust440,lvcust441
fcs67	hdisk1035,1037,1039,1041,1043	33.82	lvstk439,lvstk866,lvstk867,lvstk868,lvstk869,lvstk870,lvstk871,lvstk872,lvstk873
fcs67	hdisk1045,1047,1049,1051,1053	33.82	lvstk874,lvstk875,lvstk876,lvstk877,lvstk878,lvstk879,lvstk880,lvstk881,lvstk882
fcs68	hdisk396,1036,1040,1042,1044	33.82	lvstk883,lvstk884,lvstk885,lvstk886,lvstk887,lvstk888,lvstk889,lvstk890,lvstk891
fcs68	hdisk1026,1028,1030,1032,1034	33.82	lvcust442,lvcust443,lvcust444,lvcust445,lvcust446,lvcust447,lvcust448,lvcust449,lvcust450
fcs68	hdisk1046,1048,1050,1052,1054	33.82	lvstk892,lvstk893,lvstk894,lvstk895,lvstk896,lvstk897,lvstk898,lvstk899,lvstk900
fcs69	hdisk959	67.64	lvi2or125,lvi2or126
fcs69	hdisk961,963,965,967,969	33.82	lvcust451,lvcust452,lvcust453,lvcust454,lvcust455,lvcust456,lvcust457,lvcust458,lvcust459
fcs69	hdisk971,973,975,977,979	33.82	lvstk901,lvstk902,lvstk903,lvstk904,lvstk905,lvstk906,lvstk907,lvstk908,lvstk909
fcs69	hdisk981,983,985,987,989	33.82	lvstk910,lvstk911,lvstk912,lvstk913,lvstk914,lvstk915,lvstk916,lvstk917,lvstk918
fcs70	hdisk1056	67.64	lvi1cu9,lvistk9,lvistk18,lvi2cu25,lvi2cu26,lvi2cu27
fcs70	hdisk1058,1060,1062,1064,1066	33.82	lvordl1,lvordl2,lvordl3,lvordl4,lvordl5,lvordl6,lvordl7,lvordl8,lvordl9,lvordl10,lvordl11,lvnord7,lvnord8,lvnord9
fcs70	hdisk1068,1070,1072,1074,1076	33.82	lvstk583,lvstk938,lvstk939,lvstk940,lvstk941,lvstk942,lvstk943,lvstk944,lvstk945
fcs70	hdisk1078,1080,1082,1084,1086	33.82	lvstk946,lvstk948,lvstk949,lvstk950,lvstk951,lvstk952,lvstk953,lvstk954,lvstk947
fcs71	hdisk1055	67.64	lvi1cu1,lvi2cu2,lvi2cu3,lvistk1,lvistk10,lvi2cu1
fcs71	hdisk1057,1059,1061,1063,1065	33.82	lvordl12,lvordl13,lvordl14,lvordl15,lvordl16,lvordl17,lvordl18,lvordl19,lvordl20,lvordl21,lvordl22
fcs71	hdisk1067,1069,1071,1073,1075	33.82	lvstk955,lvstk559,lvstk957,lvstk958,lvstk959,lvstk960,lvstk961,lvstk962,lvstk963
fcs71	hdisk1077,1079,1081,1083,1085	33.82	lvstk964,lvstk965,lvstk966,lvstk967,lvstk968,lvstk969,lvstk970,lvstk971,lvstk972
fcs72	hdisk1087	67.64	lvroll1,lvroll2,lvroll3
fcs72	hdisk1089,1091,1093,1095,1097	33.82	lvordl23,lvordl24,lvordl25,lvordl26,lvordl27,lvordl28,lvordl29,lvordl30,lvordl31,lvordl32,lvordl33
fcs72	hdisk1099,1101,1103,1105,1107	33.82	lvstk973,lvstk974,lvstk975,lvstk976,lvstk977,lvstk978,lvstk979,lvstk980,lvstk981
fcs72	hdisk1109,1111,1113,1115,1117	33.82	lvstk982,lvstk983,lvstk984,lvstk985,lvstk986,lvstk987,lvstk988,lvstk989,lvstk990
fcs73	hdisk1088	67.64	lvroll4,lvroll5
fcs73	hdisk1090,1092,1094,1096,1098	33.82	lvordl34,lvordl35,lvordl36,lvordl37,lvordl38,lvordl39,lvordl40,lvordl41,lvordl42,lvordl43,lvordl44
fcs73	hdisk1100,1102,1104,1106,1108	33.82	lvstk991,lvstk992,lvstk993,lvstk994,lvstk995,lvstk996,lvstk997,lvstk998,lvstk999
fcs73	hdisk1110,1112,1114,1116,1118	33.82	lvstk1000,lvstk1001,lvstk1002,lvstk1003,lvstk1004,lvstk1005,lvstk1006,lvstk105,lvstk1008
fcs74	hdisk865,867,869,871,873	33.82	lvordl45,lvordl46,lvordl47,lvordl48,lvordl49,lvordl50,lvordl51,lvordl52,lvordl53,lvordl54,lvordl55
fcs74	hdisk875,877,879,881,883	33.82	lvstk1009,lvstk1010,lvstk1011,lvstk1012,lvstk1013,lvstk1014,lvstk1015
fcs75	hdisk1119	67.64	lvi2cu28,lvi2cu29,lvi2cu30

fcs75	hdisk1121,1123,1125,1127,1129		33.82	lvordl56,lvordl57,lvordl58,lvordl59,lvordl60,lvordl61,lvordl62,lvordl63,lvordl64,lvordl65,lvordl66,lvordl199,lvordl200
fcs76	hdisk1120		67.64	lvi2cu28,lvi2cu29,lvi2cu30
fcs76	hdisk1122,1124,1126,1128,1130		33.82	lvordl210,lvordl68,lvordl69,lvordl70,lvordl71,lvordl72,lvordl73,lvordl74,lvordl75,lvordl76,lvordl77,lvordl67
fcs76	hdisk1132,1134,1136,1138,1140		33.82	lvordl211,lvordl79,lvordl80,lvordl81,lvordl82,lvordl83,lvordl84,lvordl85,lvordl86,lvordl87,lvordl88,lvordl78
fcs76	hdisk1142,1144,1146,1148,1150		33.82	lvordl212,lvordl90,lvordl91,lvordl92,lvordl93,lvordl94,lvordl95,lvordl96,lvordl97,lvordl98,lvordl99,lvordl89
fcs77	hdisk1153,1155,1157,1159,1161		33.82	lvordl100,lvordl101,lvordl102,lvordl103,lvordl104,lvordl105,lvordl106,lvordl107,lvordl108,lvordl109,lvordl110
fcs77	hdisk1163,1165,1167,1169,1171		33.82	lvordl111,lvordl112,lvordl113,lvordl114,lvordl115,lvordl116,lvordl117,lvordl133,lvordl119,lvordl122,lvordl121
fcs77	hdisk1173,1175,1177,1179,1181		33.82	lvordl120,lvordl123,lvordl124,lvordl125,lvordl126,lvordl127,lvordl128,lvordl129,lvordl130,lvordl131,lvordl132
fcs78	hdisk1154,1156,1158,1160,1162		33.82	lvordl118,lvordl134,lvordl135,lvordl136,lvordl137,lvordl155,lvordl144,lvordl140,lvordl141,lvordl142,lvordl143,lvordl202,lvordl203
fcs78	hdisk1164,1166,1168,1170,1172		33.82	lvordl139,lvordl145,lvordl146,lvordl147,lvordl148,lvordl149,lvordl150,lvordl151,lvordl152,lvordl153,lvordl154
fcs78	hdisk1174,1176,1178,1180,1182		33.82	lvordl138,lvordl156,lvordl157,lvordl158,lvordl159,lvordl160,lvordl161,lvordl162,lvordl163,lvordl164,lvordl165
fcs79	hdisk896		67.64	lvi2or121,lvi2or122,lvi2or123,lvi2or124
fcs79	hdisk898,900,902,904,906		33.82	lvordl166,lvordl167,lvordl168,lvordl169,lvordl170,lvordl171,lvordl172,lvordl173,lvordl174,lvordl175,lvordl176,lvhist1
fcs79	hdisk908,910,912,914,916		33.82	lvordl177,lvordl178,lvordl179,lvordl180,lvordl181,lvordl182,lvordl183,lvordl184,lvordl185,lvordl186,lvordl187
fcs79	hdisk918,920,922,924,926		33.82	lvordl188,lvordl189,lvordl190,lvordl191,lvordl192,lvordl193,lvordl194,lvordl195,lvordl196,lvordl197,lvordl198
ssa0	hdisk290,291	218GB raid_5		tpc_oldlogvg
ssa1	hdisk292,293	218GB raid_5		tpc_oldlogvg
ssa2	hdisk295,296	218GB raid_5		tpc_oldlogvg
ssa4	hdisk393,397	218GB raid_5		tpc_oldlogvg
ssa5	hdisk398,399	218GB raid_5		tpc_oldlogvg
ssa7	hdisk496,497	218GB raid_5		tpc_oldlogvg

5.3 Data Base Model Implemented

A statement must be provided that describes the data base model implemented by the DBMS used.

The database manager used for this testing was Oracle Database 10g Enterprise Edition from Oracle Corp. Oracle Database 10g Enterprise Edition is a relational DBMS.

5.4 Partitions/Replications Mapping

The mapping of data base partitions/replications must be explictly described.

IBM did not implement horizontal or vertical partitioning for this TPC-C test.

5.5 60 day space calculations

IBM eServer pSeries 690 Turbo Model 7040-681

TPM 768839
Warehouses 60768

SEGMENT	TYPE	TSPACE	BLOCKS	BLOCK SIZE	KBYTES	FIVE_PCT(BYTES)	DAILY_GROW	TOTAL
CUSTCLUSTER	CLUSTER	CUST	390827008	4096	1563308032	78165402	0	1641473433.60
D2CLUSTER	CLUSTER	WARE	61440	2048	122880	6144	0	129024.00
DISTCLUSTER	CLUSTER	WARE	327424	4096	1309696	65485	0	1375180.80
HIST	TABLE	HIST	17187072	8192	137496576	0	27833792	165330368.00
ICUSTOMER	INDEX	ICUST1	11790080	4096	47160320	2358016	0	49518336.00
ICUSTOMER2	INDEX	ICUST2	25132800	4096	100531200	5026560	0	105557760.00
IDISTRICT	INDEX	WARE	6848	2048	13696	685	0	14380.80
IITEM	INDEX	ITEMS	950	2048	1900	95	0	1995.00
IORDR2	INDEX	IORD2	18388480	4096	73553920	3677696	0	77231616.00
ISTOCK	INDEX	ISTR	35873280	4096	143493120	7174656	0	150667776.00
ITEMCLUSTER	CLUSTER	ITEMS	7168	2048	14336	717	0	15052.80
IWARE	INDEX	WARE	544	2048	1088	54	0	1142.40
NORDCLUSTER	CLUSTER	NORD	4539392	4096	18157568	907878	0	19065446.40
ORDRCLUSTER	CLUSTER	ORD	118518336	16384	1896293376	94814669	0	1991108044.80
STOKCLUSTER	CLUSTER	STOCKS	591788032	4096	2367152128	118357606	0	2485509734.40
SYSTEM	SYS	SYSTEM	229122	4096	916488	0	0	916488.00
WARE2CLUSTE	CLUSTER	WARE	65536	2048	131072	6554	0	137625.60
WARECLUSTER	CLUSTER	WARE	33024	4096	132096	6605	0	138700.80
Total Bytes			1,214,776,536		6,349,789,492	310,568,821	27,833,792	6,688,192,105.40

Dynamic space in Kbytes 137,496,576
Static space in Kbytes 6,522,861,737
Free space in Kbytes 27,833,792

Daily growth in Kbytes 27,833,792
Daily spread 0 Oracle may be configured such that daily spread is 0
60-day space (in Kbytes) 8,192,889,257

60-day (GB) 7,813.35

new_order 136287389
Log block size 512 Redo blocks written 1350365574
Log blocks/tpmC 9.91 Number of log blocks used in one tpmC
8-hour log (GB) 1,743.58

Disk Type	Disk Formatted Capacity	SUT # of disks	SUT Capacity(GB)	Priced # of disks	Priced Capacity(GB)	Space usage (GB)	
						DB	35849.53
						RAID	3499.20
						OS	50.91
18.2	17316	2120	35849.53	2120	35849.53	Total Space	39399.64
RAID(8-36.4GB)	298598.4	12	3,499.20	12	3,499.20		

6. Clause 5: Performance Metrics and Response Time Related Items

6.1 Response Times

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

Table 6-1 list the response times and the ninetieth percentiles for each of the transaction types for the measured system.

6.2 Keying and Think Times

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

Table 6-1 list the TPC-C keying and think times for the measured system.

Table 6-1. IBM eServer pSeries 690 Turbo Model 7040-681 Response, Think and Keying Times

Response Times	New Order	Payment	Order Status	Delivery (int./def.)	Stock Level	Menus
90 %	0.64	0.63	0.63	0.19/0.05	0.63	0.01
Average	0.31	0.3	0.3	0.11/0.03	0.3	0.01
Maximum	25.31	24.46	21.96	21.00/4.94	23.03	2.53
			Think Times			
Minimum	0.01	0.01	0.01	0.01	0.01	N/A
Average	12.02	12.02	10.01	5.02	5.02	N/A
Maximum	120.20	120.20	100.10	50.20	50.20	N/A
			Keying Times			
Minimum	18.00	3.00	2.00	2.00	2.00	N/A
Average	18.01	3.01	2.01	2.01	2.01	N/A
Maximum	18.11	3.12	2.09	2.09	2.10	N/A

6.3 Response Time Frequency Distribution

Response time frequency distribution curves must be reported for each transaction type.

Figure 6-3-1. IBM eServer pSeries 690 Turbo Model 7040-681 New-Order Response Time Distribution

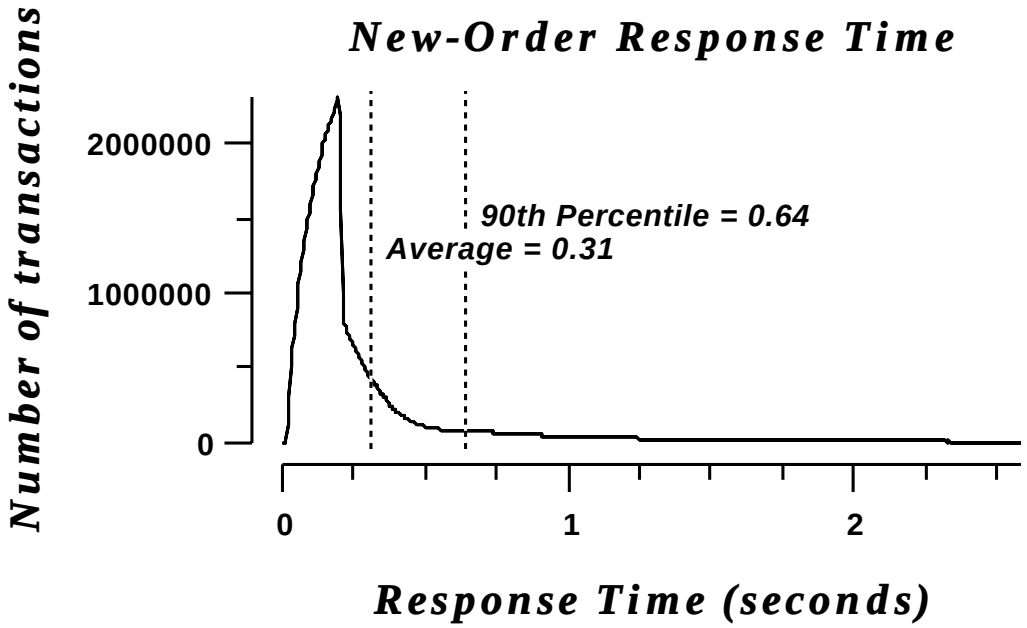


Figure 6-3-2. IBM eServer pSeries 690 Turbo Model 7040-681 Payment Response Time Distribution

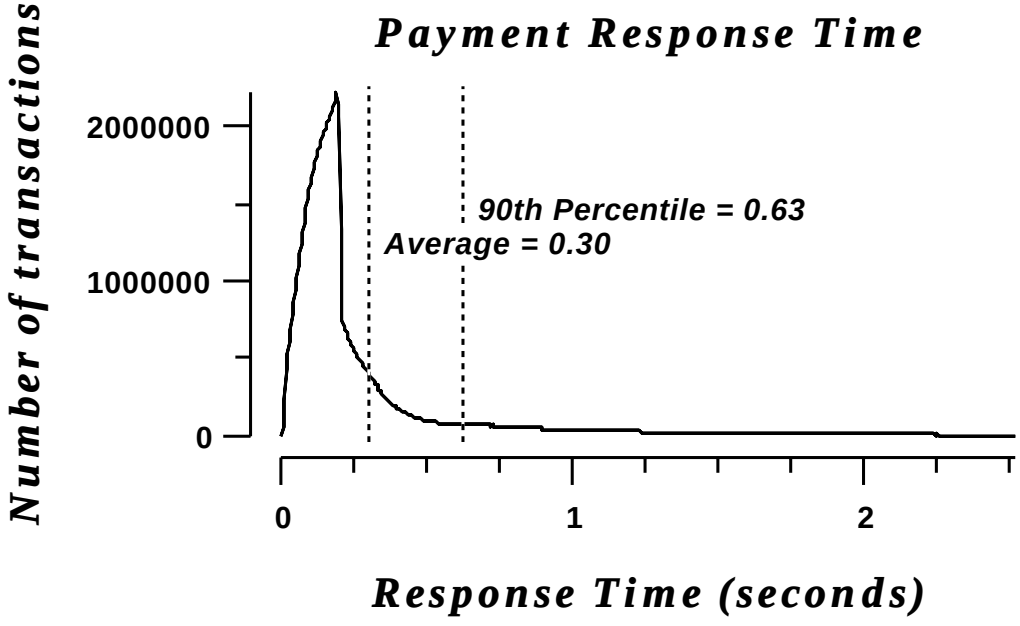


Figure 6-3-3. IBM eServer pSeries 690 Turbo Model 7040-681 Order-Status Response Time Distribution

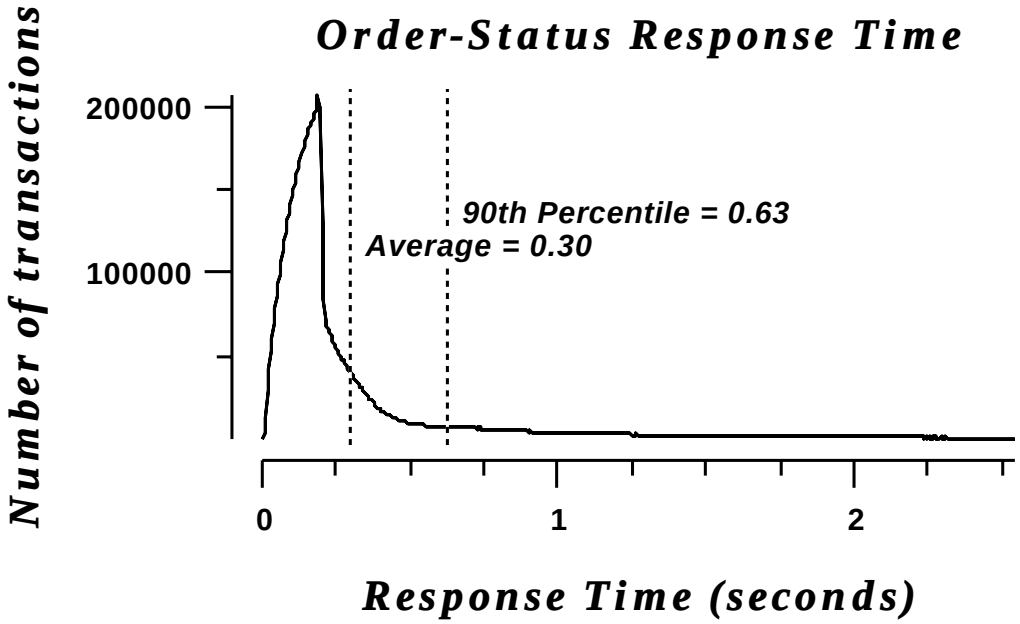


Figure 6-3-4. IBM eServer pSeries 690 Turbo Model 7040-681 Delivery (Interactive) Response Time Distribution

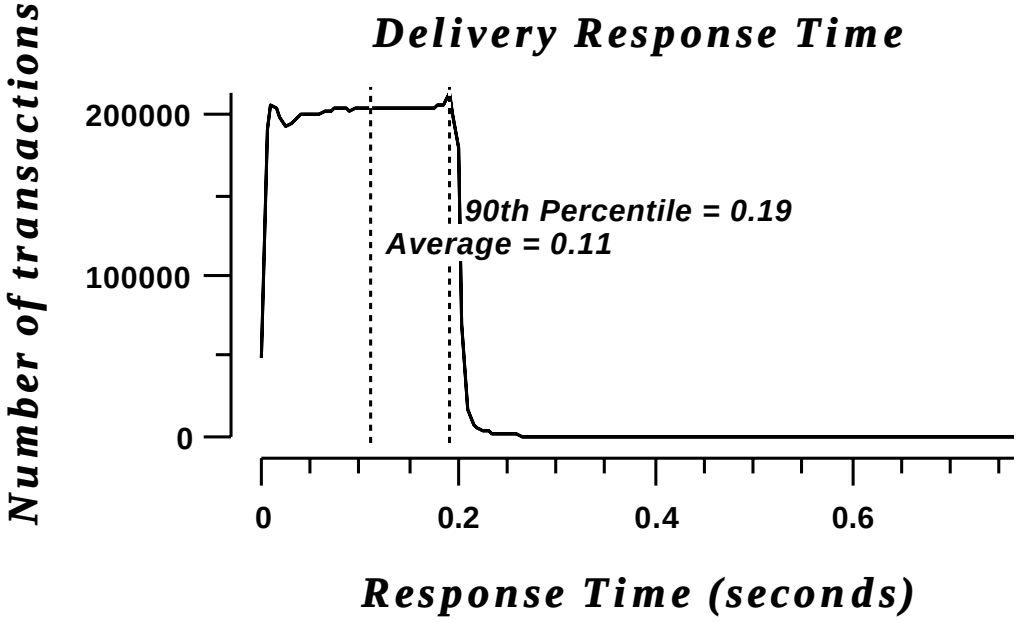


Figure 6-3-5. IBM eServer pSeries 690 Turbo Model 7040-681 Delivery (Deferred) Response Time Distribution

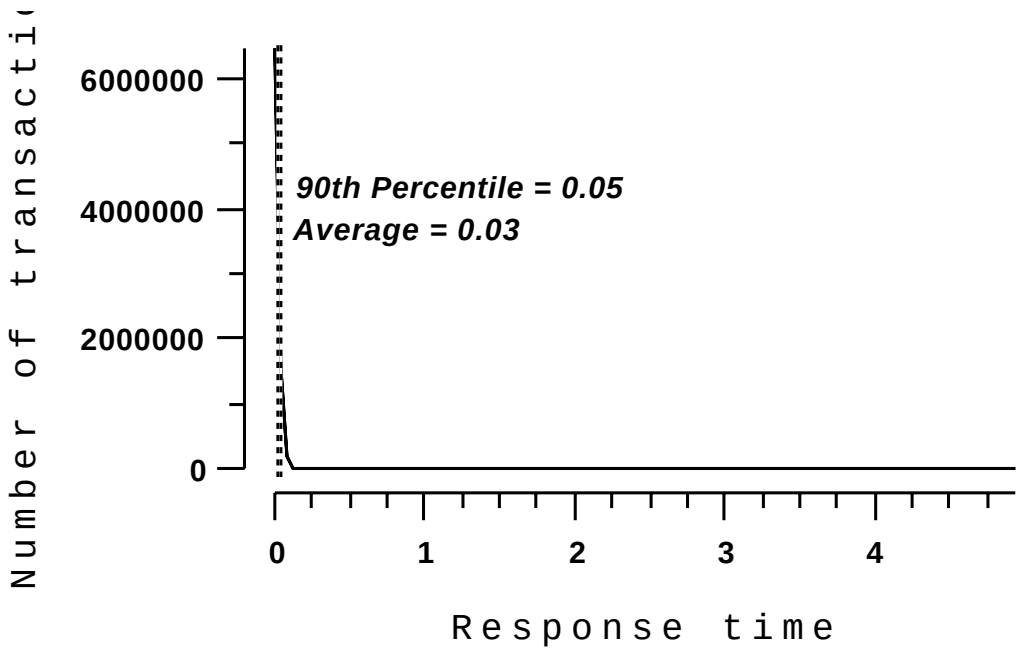
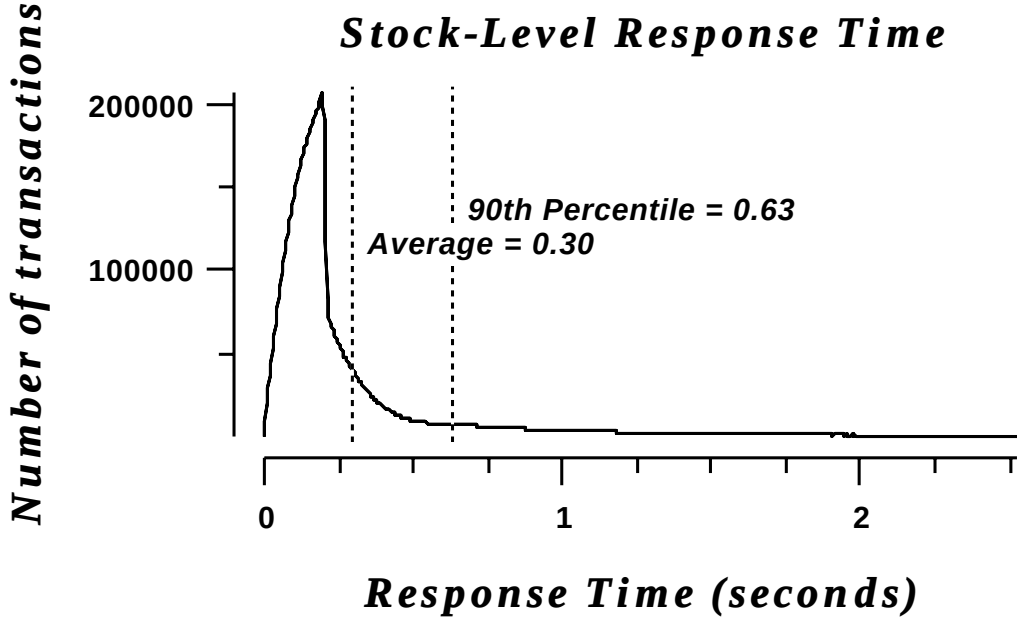


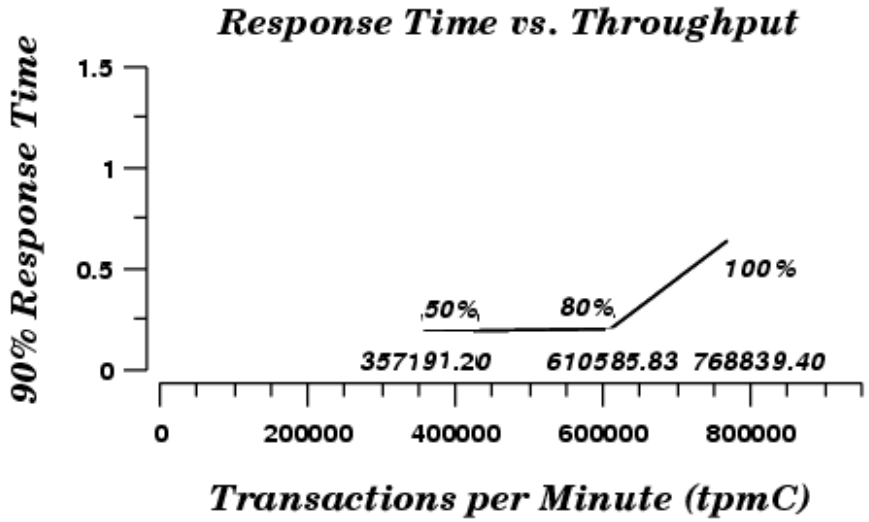
Figure 6-3-6. IBM eServer pSeries 690 Turbo Model 7040-681 Stock Level Response Time Distribution



6.4 Performance Curve for Response Time versus Throughput

The performance curve for response times versus throughput must be reported for the New-Order transaction.

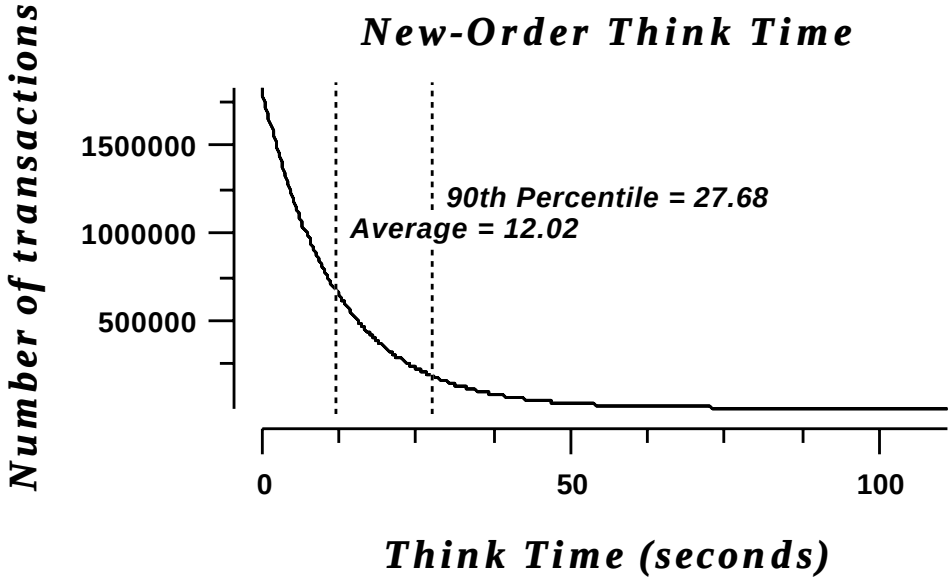
Figure 6-4-1. IBM eServer pSeries 690 Turbo Model 7040-681 New-Order Response Time vs. Throughput



6.5 Think Time Frequency Distribution

A graph of the think time frequency distribution must be reported for the New-Order transaction.

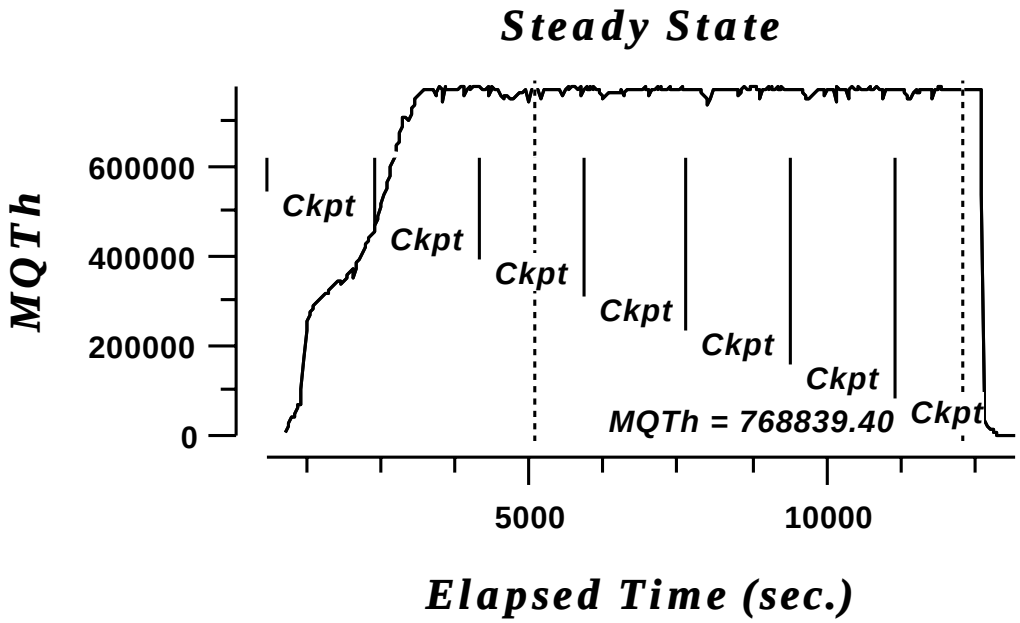
Figure 6-5-1. IBM eServer pSeries 690 Turbo Model 7040-681 New-Order Think Time Distribution



6.6 Throughput versus Elapsed Time

A graph of throughput versus elapsed time must be reported for the New-Order transaction.

Figure 6-6-1. New-Order Throughput vs. Elapsed Time



6.7 Steady State Determination

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval must be described.

All the emulated users were allowed to logon and do transactions. The time stamping interval was set to start after several minutes of rampup. Refer to the Numerical Quantities Summary pages for the rampup time. Figure 6.6.1 New-Order throughput versus Elapsed Time graph shows that the system was in steady state at the beginning of the Measurement Interval.

6.8 Work Performed During Steady State

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

6.8.1 Transaction Flow

For each of the TPC Benchmark™C transaction types, the following steps are executed:

IBM TXSeries Developer for AIX V5, Encina interface, was used as a transaction manager (TM). Each transaction was divided into three programs. The front end program handled all screen I/O, a database client program which connected to the database and served as a Encina Server (a back end program), and a database server program which handled all database operations at the SUT. Both the front end and back end programs ran on the client system. The front end program communicates with database client program through DCE RPCs. The database client program communicates with the Server system over Ethernet using SQL*Net calls. Besides calling Encina initialization code during startup, all other functions are transparent to the application code. Encina routes the transaction and balances the load according to the options defined in the configuration file in appendix B.2. Encina Universal Binding is used to tie an Encina client thread to a particular Encina Application Server. The transaction flow is described below.

- Each client machine is a node in an Encina Cell.

- Two types of servers are configured in each node: one processes the delivery transactions and one handles all other transactions.
- The delivery server is configured with one processing agent with 2 server manager DCE threads, and 3 background threads to process deferred deliveries. Each background thread has one connection to the database.
- The other 24 servers are configured each with just one processing agent. Each processing agent has 1 server manager DCE threads. Each thread has one connection to the database.
- When the Encina clients are started, they connect to the Encina cell.
- When terminals are started, each terminal connects to the Encina client. The client spawns a thread for each connection to handle that connection. The thread executes the 'process_terminal' routine. The process_terminal displays the TPC-C transaction menu on the user terminal.
- The TPC-C user chooses the transaction type and proceeds to fill the screen fields required for transaction.
- The process_terminal accepts all values entered by the user and transmits those values to one of the TPC_C back end programs. The transaction is performed through a DCE RPC. There is an interface for each TPC-C transaction type and each TPC-C back end program exports one or more of these interfaces. (The delivery servers export only the delivery interface, the other servers export the other four interfaces, and only those). Encina transparently routes the RPC to one of the servers exporting the corresponding interface.
- A TPC-C back end server program receives a RPC and proceeds to execute all database operations related to the request. All information entered on the user terminal is contained in the RPC.
- Once the transaction is committed, the server program fills in the output parameters. The RPC is then sent back to the client program.
- When the RPC returns to the client, the process_terminal routine writes the transaction out on the user terminal.

6.8.2 Database Transaction

All database operations are performed by the TPC-C back-end programs. The process is described below:

Using SQL*Net calls, the TPC-C back-end program interacts with Oracle 10g Server to perform SQL data manipulations such as update, select, delete and insert, as required by the transaction. After all database operations are performed for a transaction, the transaction is committed.

Oracle 10g Server proceeds to update the database as follows:

When Oracle 10g Server changes a database table with an update, insert, or delete operation, the change is initially made in memory, not on disk. When there is not enough space in the memory buffer to read in or write additional data pages, Oracle 10g Server will make space by flushing some modified pages to disk. Modified pages are also written to disk when a checkpoint occurs. Before a change is made to the database, it is first recorded in the transaction log. This ensures that the database can be recovered completely in the event of a failure. Using the transaction log, transactions that started but did not complete prior to a failure can be undone, and transactions recorded as complete in the transaction log but not yet written to disk can be redone.

6.8.3 Checkpoints

A checkpoint is the process of writing all modified data pages to disk. The TPC-C benchmark was setup to automatically checkpoint every 30 minutes. Two checkpoints occurs during the rampup period, and four other occurring during the measurement interval. The incremental checkpoint duration during the measurement is 29 minutes and 1 second.

6.9 Measurement Interval

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

A two hour Measurement Interval was used. Further, the measurement interval is a multiple of the checkpoint interval. This demonstrates that a different measurement interval over the eight hour period would yield similar throughput results. No connections were lost during the run.

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

7.1 RTE Availability

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

IBM used an internally developed RTE for these tests. Appendix D contains the scripts used in the testing.

7.2 Functionality and Performance of Emulated Components

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

In the benchmark configuration the Remote Terminal Emulator (RTE) communicates with the client system over Ethernet. The 60 eServer pSeries 6F1 emulates a network of 607,680 RS/6000 Model 44P-170 workstations. The communications mechanism used in the benchmark and priced configurations are the same. In the benchmark configuration a separate Ethernet LAN was used to connect two driver systems to a eServer pSeries Model 6E4 client system. In other words, there were several separate LAN segments between every one driver to one client. Each LAN segment in the priced configuration is used to connect 844 workstations.

7.3 Network Bandwidth

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

The Ethernet used in the LAN complies with the IEEE 802.3 standard and has a bandwidth of 10 Megabits per second Half Duplex. Each LAN segment in the IBM eServer pSeries 690 Turbo Model 7040-681 configuration connected 844 workstations.

7.4 Operator Intervention

If the configuration requires operator intervention, the mechanism and the frequency of this intervention must be disclosed.

No operator intervention is required to sustain the reported throughput during the eight hour period.

8. Clause 7: Pricing Related Items

8.1 Hardware and Programs Used

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

The detailed list of all hardware and programs for the priced configuration is listed in the pricing sheets (please refer to Section 8.2 for details) for each system reported. The prices for all products and features that are provided by IBM are available the same day as product or feature availability.

Pricing for TXSeries Developer for AIX V5 includes 3 years of support. Customers should call for order details.

8.2 Three Year Cost of System Configuration

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.

The price sheets for the eServer pSeries 690 Turbo Model 7040-681 are contained on the following page. The basis for the discounts used are:

3-year Term Maintenance Contract Discount

This discount is available for customers who sign a 3-year maintenance agreement on the hardware. A discount of 3% is available for customers when they sign a 3-year maintenance agreement.

Scope Incentive

A 2% discount is applied for the ServiceElect contract that combines hardware maintenance with one or more services, which in this pricing report the selected service is SupportLine.

3-year Maintenance Prepay Discount

This is a discount for prepayment of maintenance costs. A discount of 10.36% is available for this configuration based on payment for three years maintenance at the time of purchase. This discount is applied to the balance after the 3-year term maintenance contract discount and Scope discounts are applied.

Hardware Volume Revenue Discount

The IBM-supplied hardware and software is discounted by 43% from the list price, based on the dollar value of this configuration only.

8.3 Availability Dates

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

All products are generally available today except the following:

Product	Availability Date
AIX APAR IY48265 V5.2	February 29, 2004
TXSeries Developer for AIX V5	December 1, 2003
Oracle Database 10g Enterprise Edition	December 31, 2003

8.4 Statement of tpmC and Price/Performance

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

.

System	tpmC	3-year System Cost	$/tpmC	Availability Date
IBM eServer pSeries 690 Turbo Model 7040-681	**768,839.40**	**$6,574,014**	**8.55$/tpmC**	All HS/SW available as shown in Section 8.3

Description	Part No.	Source	Unit Price	Qty	Ext Price	Maint Price
Server Hardware						
IBM eServer pSeries 690 (CD, cables, clock card)	7040-681	1	13,394	1	13,394	20,760
128MB (4x32) L3 Cache, 1700MHz 8w MCM	4199	1	375,000	4	1,500,000	372,000
64GB H-Memory (inw ard and outw ard 4488,4489)	4488	1	344,000	4	1,376,000	
Std CEC Fan Pw r cbl grp, service proc. 6161, 6162	6161	1	600	1	600	
Standard CEC DC/DC Converter Assembly	6170	1	4,700	2	9,400	
Pow er Cable Group, 1 thru 4 proc. module	6181	1	720	1	720	
Additional DC/DC Converter Assembly, (DCA)	6189	1	4,200	3	12,600	
Capacitor book (for 2 MCMs)	6198	1	1,800	2	3,600	
Service Processor + Dual Rio Loop (GX), I/O loop adapter	6418	1	15,000	1	15,000	
Backplane, Central Electronics Complex	6565	1	50,000	1	50,000	
Attachment Cable, Hardw are Management Console	8121	1	75	1	75	
Media Drw r, 1U, Op Pnl/Dskt/4 SCSI Media Bays	8692	1	1,700	1	1,700	
I/O Draw er	7040-61D	1	3,980	8	31,840	32,640
Converter Cable, VHDCI to P, 0.3M	2118	1	60	8	480	
SCSI cable - B&C to Media Draw er	2122	1	275	1	275	
RIO-G Cable, 500Hz, 0.5m, 2*2m (3145, 3149)	3145	1	1010	8	8,080	
36.4 GB 10,000 RPM Ultra3 SCSI Disk Drive	3158	1	1,283	3	3,849	
IBM Gigabit Ethernet-SX PCI-X Adapter	5700	1	1,700	1	1,700	
Cable Grp, 4xUPIC/2xRIO, BPM(IO#1 thru 4) (6121-6129)	6121	1	3,200	1	3,200	
I/O Drw r DC/DC Converter assembly, (DCA)	6172	1	4,000	16	64,000	
Pow er Cable, B&C to Media Draw er 6179,SCSI adapter 620	6179	1	695	1	695	
Advanced Serial RAID Adapter, 32MB Cache (6230, 6235)	6230	1	3,575	6	21,450	
2 Gigabit Fibre Channel PCI-X Adapter	6239	1	2,582	66	170,412	
Ultra3 SCSI 4-Pack	6564	1	500	3	1,500	
B&C Planar, 10 PCI Slots, 2x integrated SCSI	6571	1	8,000	16	128,000	
Rack, Front doors, Back doors	7040-61R	1	10,550	1	10,550	3,960
Bulk Pow er Regulator (BPR)	6186	1	4,000	4	16,000	
Bulk Pow er Controller (BPC), 4 Fans + 3 DCAs	6187	1	1,900	2	3,800	
Bulk Pow er Distribution (BPD), 10 DCAs	6188	1	3,500	4	14,000	
Line Cord, 60A, 14', IEC309 Plug, Chargeable	8678	1	1,000	2	2,000	
Regatta-H Bulk Pow er Module (2x BPA, EPO PNL)	8690	1	5,000	1	5,000	
Expansion Rack 24", 42 EIA	8691	1	8,000	1	8,000	
System Console, serial attach. cable 6M	7316	1	4,400	1	4,400	
IBM T541H 15 inches TFT Color Monitor, keyboard, graphics	3637	1	974	1	974	
				Subtotal	**3,483,294**	**429,360**
Storage						
System Rack Model T42 (6081,6098)	7014-T42	1	4,270	1	4,270	888
Additional Pow er Distribution Unit	6171	1	1,000	1	1,000	
				Subtotal	**5,270**	**888**
SSA Disk Subsystem,50/60 Hz AC, pw r supply, cover, cabl	7133-D40	1	15,125	6	90,750	26,496
Advanced 10K rpm/36.4 GB Disk Module	8536	1	5,900	96	566,400	
				Subtotal	**657,150**	**26,496**
Racked FAStT Storage Solution	2101-200	1	4,850	16	77,600	12,288
FAStT EXP 700	1740-1RU	1	6,000	160	960,000	230,400
Short Wave SFP	2210	1	499	320	159,680	
18.2 GB 15K drive	5211	1	681	2,120	1,443,720	
FAStT700 Storage Server	1742-1RU	1	46,499	32	1,487,968	46,080
Short Wave SFP GBIC	2210	1	499	128	63,872	
				Subtotal	**4,192,840**	**288,768**
Server Softw are						
AIX 5L v5.2 (media only)	5765-E61	1	50	1	50	
AIX SWMA Subsription	5771-SUB	1	247	96	23,712	
AIX SWMA Support	5771-SPT	1	697	96	0	66,912
IBM C for AIX V5	D5A1ELL	1	559	1	559	104
Oracle Database 10g Enterprise Edition, Per Processor, Unlimited Users, for 3 years		2	20,000	32	640,000	
Oracle Database Server Support Package for 3 years		2				6,000
				Subtotal	**664,321**	**73,016**
Client Hardw are and Softw are						
pSeries 630-6E4, CD-ROM, 18.2 GB Disk, Pw r Supply	7028-6E4	1	3,941	30	118,230	90,000
FC IExp Config (420C/420E) AIX	8037	1	23,317	30	699,510	
T541H 15" TFT Color Monitor, Captured Cable, keyboard	3637	1	562	1	562	
POWER GXT135P Graphics Accelerator -Digital support	2849	1	412	1	412	
SCSI Connector Cable + 4-Pack How Sw ap Back Plane	4254	1	354	30	10,620	
2048MB (4x512MB) SDRAM DIMM Memory,Express	8080	1	0	120	0	
4-port 10/100 Mbps Ethernet adapter	4961	1	1,250	90	112,500	
2-w ay 1.2 GHz POWER4 processor card, Express	8107	1	0	60	0	
AIX SWMA Subsription	5773-SUB	1	157	240	37,680	
AIX SWMA Support	5773-SPT	1	427	240	0	102,480
AIX 5L version 5	5692-A5L	1	50	30	1,500	
TXSeries Developer for AIX V5, per proc, w/3 yrs maint.		1	2,599	120	311,904	
				Subtotal	**1,292,918**	**192,480**
Third Party Hardw are						
WideBand Series Fast Ethernet/Gigabit Sw itch	WB24T2EML3	3	1,449	4	5,796	
				Subtotal	**5,796**	
Oracle Mandatory E-Business Discount on License and Support		2	(129,200)	1	(129,200)	
				Discounts	**(4,211,211)**	**(398,172)**
				Total	**5,961,178**	**612,836**

Audited by: Francois Raab, InfoSizing (w w w .infosizing.com)

Pricing Sources:

1=IBM: Bill Casey, eServer pSeries Offering Manager, w rcasey@us.ibm.com, 512-838-1422

2=Oracle: MaryBeth Pierantoni, mary.beth.pierantoni@oracle.com, 650-506-2118

3=WideBand Corp.

Three-Year Cost of Ownership	**$6,574,014**
tpmC	**768,839.40**
$/tpmC	**8.55**

9. Clause 9: Audit Related Items

If the benchmark has been independently audited, then the auditor's name, address, phone number, and a brief audit summary report indicating compliance must be included in the Full Disclosure Report. A statement should be included, specifying when the complete audit report will become available and who to contact in order to obtain a copy.

The auditor's attestation letter is included in this section of this report

INFOSIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL CERTIFIED AUDITOR

Sponsors:

John J. Makis
IBM eServer Performance
11501 Burnet Road
Austin, Texas 78758

Vineet Buch
Oracle Corporation
100 Oracle Parkway
Redwood Shores, CA 94065

September 10, 2003

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

Platform: IBM eServer pSeries 690 Turbo Model 7040-681 c/s
Operating system: AIX 5L V5.2
Database Manager: Oracle Database 10g Enterprise Edition
Transaction Manager: IBM TXSeries Developer for AIX V5

The results were:

CPU's Speed	Memory	Disks	NewOrder Response Time - 90%	tpmC
Server: IBM eServer pSeries 690 Turbo				
32 x POWER4+ (1.7 GHz)	512 GB (128 MB L3 cache/cpu)	3 x 18.2GB SCSI 96 x 36.4GB SSA 2120 x 18.2GB FAStT	0.64 Sec.	768,839.4
Thirty (30) Clients: IBM eServers pSeries 7028-6E4 (Specification for each)				
4 x RS64 IV (1.2 GHz)	8 GB (16 MB L3 cache/cpu)	1 x 18.2 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.

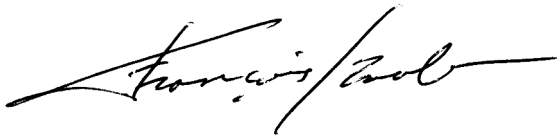
The following verification items were given special attention:

- 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 incremental 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:

None.

Respectfully Yours,



François Raab, President

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

Appendix A: TPC-C Application Source

A.1 Client/Terminal Handler code

callora.c

```
/*
 * callora.c
 *
 * $Revision: 1.3 $
 * $Date: 1999/05/06 21:28:29 $
 * $Log:          callora.c,v $
 *
 * $TALog: callora.c,v $
 * Revision 1.3  1999/05/06  21:28:29  oz
 * - Removed all the .. from the includes
 * - Added -I.. to the  makefiles instead
 * - Moved all the thread related code and connection
 *   selection to serverMon.c
 *
 * - get_db_ready now does not take the number of connections
 * - Export create_connection() and clean_connection(void *)
 * - All the transactions take a connection pointer as a first param
 * [from r1.2 by delta oz-24309-TPCC-add-oracle8.1-code, r1.5]
 *
 * Revision 1.2  1999/04/19  20:14:48  oz
 * - Moved all the simulated code to server.c
 * - Created nulldb.c for compilation with no DB
 * [from r1.1 by delta oz-24331-TPCC-move-sim-code-to-common-file,r1.1]
 *
 * Revision 1.1  1999/04/19  14:37:27  oz
 * [added by delta wenjian-23742-TPCC-update-with-Raliegh-code,r1.3]
 *
 * Revision 1.15  1998/10/22  20:51:00  wenjian
 * [merge of changes from 1.6 to 1.14 into 1.12]
 *
 * Revision 1.14  1998/10/08  14:17:57  dongfeng
 * Add codes for doing web-based tpcc.
 * [from r1.6 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.1]
 *
 * Revision 1.12  1998/09/04  19:17:54  wenjian
 * Add new variables: more_srv_work, period_to_add_rt, and
 * period_to_check_tran to replace the original constants in
 * order to control the increment of server RT.
 * [from r1.11 by delta wenjian-23183-TPCC-sync-AIX-code-with-Austin,r1.5]
 *
 * Revision 1.11  1998/08/28  18:29:56  wenjian
 * This delta sync the TPCC code with Austin.
 *
 * Modify get_wait_time():
 *  - add rt_increment so that the wait time is increased in a certain time;
 *  - rt_increment is reset to 0 at the beginning of each run
 *  - the waiting time is different for differenttran type.
 * [from r1.8 by delta wenjian-23183-TPCC-sync-AIX-code-with-Austin,r1.1]
 *
 * Revision 1.8  1998/08/18  14:38:37  wenjian
 * Change the wait time for NewOrder to 0.23 second
 * [from r1.6 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.4]
 *
 * Revision 1.6  1998/06/17  15:28:50  wenjian
 * - Reduce matrix size
 * - In get_wait_time(), the waiting time is decided by transaction type.
 * [from r1.5 by delta wenjian-22495-TPCC-add-new-feature-to-monitor-tpcc-clients, r1.2]
 *
 * Revision 1.5  1998/02/17  22:06:58  wenjian
 * Define macro RANDOM as rand on NT and random on other platforms
 * [from r1.4 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.1]
 *
 * Revision 1.4  1998/01/23  15:07:42  oz
 * - Updated the SP TPCC directory to the latest files used
 *   during the SP tpcc audit.
 * [from r1.3 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
 *
 * Revision 1.1  1997/07/22  21:17:14  radha
 * [added by delta radha-20360-TPCC-integrate-with-Oracle-7322-drivers, r1.1]
 *
 *
 */


#include <stdio.h>
#include <time.h>
#include <string.h>
#include "serverDebug.h"

#ifdef MULTIPLE_INTERFACE
#include "common/neworder.h"
#include "common/payment.h"
#include "common/stocklevel.h"
#include "common/orderstatus.h"
#else
#include "common/tpcc_trans.h"
#endif
#include "common/databuf.h"
#include "server.h"

#ifdef WIN32
#include <winsock.h>
#endif


#include "tpcc_info.h"

#ifdef WIN32
#define RANDOM rand
#else
#define RANDOM random
#endif

extern int server_null_test;
extern void *create_ora_connection();

#ifdef DEBUG_SERVER
#define PRINT_NEW_IN(a, b) fprintf(stderr, "%s\n", b); print_new_in(a)
#define PRINT_NEW_ORDER(a, b) fprintf(stderr, "%s\n", b); print_new_order(a)
#define PRINT_NEW_RES(rc, a) \
   fprintf(stderr, "<R do_new_order, rc=%d, transtatus=%d, duplicates=%d, all_local=%d\n",\
         rc, (a)->s_transtatus, (a)->s_all_local, (a)->duplicate_items)

#else
#define PRINT_NEW_RES(rc, a)
#define PRINT_NEW_ORDER(a, b)
#define PRINT_NEW_IN(a, b)
#define PRINT_DIST_NEW_ORDER(a, b)
#endif


#define TPCC_RET_SCP(a,b,len) \
    strncpy((char *)dataP->b, (char *)oraStruct.a, len); \
    (char *)dataP->b[(len)-1] = '\0'
#define TPCC_CP(a,b) oraStruct.a = dataP->b
#define TPCC_SCP(a,b,len) strncpy((char *)oraStruct.a, (char *)dataP->b, len)
#define TPCC_RET_CP(a,b) dataP->b = oraStruct.a

#define TPCCP_RET_SCP(a,b,len) \
    strncpy((char *)dataP->b, (char *)oraStructP->a, len); \
    dataP->b[(len)-1] = '\0'
#define TPCCP_CP(a,b) oraStructP->a = dataP->b
#define TPCCP_SCP(a,b,len) strncpy((char *)oraStructP->a, (char *)dataP->b, len)
#define TPCCP_RET_CP(a,b) dataP->b = oraStructP->a

/*
 * Talk to Oracle
 */
int get_db_ready(char *dbName, int flag)
{
   int rc;
   char dvryFileName[100];
   extern char *tpcc_serverName;

   AUDITLOG(("> get_db_ready to %s flag %d\n", dbName, flag));
   if (server_null_test) return(0);

   fprintf(stderr, ">> get_db_ready, db: %s, flag %d\n", dbName, flag);

   sprintf (dvryFileName, "/home/encina/runs/deliveries/%s",
         tpcc_serverName);
   rc = TPCinit (serverIdNumber, "tpcc", "tpcc", dvryFileName);
   err_printf("TPCinit(%d, tpcc, tpcc, %s) returned %d\n",
                              serverIdNumber, dvryFileName, rc);
   if (rc) {
      fprintf(stderr, "TPCinit(%d, tpcc, tpcc, %s) returned %d\n",
                              serverIdNumber, dvryFileName, rc);
   }

   AUDITLOG(("< get_db_ready rc %d\n", rc));
   return(rc);
}

void *create_connection() {
   return create_ora_connection();
}


void do_delivery(cnP, dataP)
    void *cnP;
    delivery_data_t *dataP;
{
   struct delstruct oraStruct;
   int rc;

   AUDITLOG(("> do_delivery\n"));

   TPCC_CP(delin.w_id, w_id);
   TPCC_CP(delin.o_carrier_id, o_carrier_id);
   TPCC_CP(delin.qtime, start_queue);
   TPCC_CP(delin.in_timing_int, queued_time);
```

```
  DPRINT(("Calling TPCdel: w_id %d, o_carrier_id %d, %f qtime, %d in_timing_int\n",
                  oraStruct.delin.w_id, oraStruct.delin.o_carrier_id,
                  oraStruct.delin.qtime, oraStruct.delin.in_timing_int));

  rc = TPCdel(cnP, &oraStruct);
  if ((rc != 0) && (rc != -666)) {
     err_printf("Error TPCdel: terror %d, rc %d, retry %d, w_id %d, o_carrier_id %d, %f qtime,
%din_timing_int\n",
                  oraStruct.delout.terror, rc, oraStruct.delout.retry,
                  oraStruct.delin.w_id, oraStruct.delin.o_carrier_id,
                  oraStruct.delin.qtime, oraStruct.delin.in_timing_int);
  }
  dataP->header.returncode = rc == 0 ? TPCC_SUCCESS : oraStruct.delout.terror;
  AUDITLOG(("< do_delivery rc %d\n", rc));
}

void copyout_order_status(orderStatus_data_t  *dataP,
                                                  struct ordstruct *oraStructP)
{
  int i;
  TPCCP_RET_CP(ordout.c_balance, c_balance);
  TPCCP_RET_CP(ordout.o_id, o_id);
  TPCCP_RET_CP(ordout.o_carrier_id, o_carrier_id);
  TPCCP_RET_CP(ordout.o_ol_cnt, o_ol_cnt);
  TPCCP_RET_CP(ordout.c_id, c_id);
#define I_CP(ind, a,b) dataP->item[ind].b = oraStructP->ordout.a[ind]
#define I_SCP(ind, a, b, len) \
 strncpy((char *)dataP->item[ind].b, (char *)oraStructP->ordout.a[ind], len); \
 dataP->item[ind].b[(len) - 1] = '\0'
  for (i=0; i<oraStructP->ordout.o_ol_cnt && i < 15; i++) {
     I_CP(i, ol_amount, ol_amount);
     I_CP(i, ol_i_id, ol_i_id);
     I_CP(i, ol_supply_w_id, ol_supply_w_id);
     I_CP(i, ol_quantity, ol_quantity);
     I_SCP(i, ol_delivery_d, delivery_date, 11);
  }
#undef I_CP
#undef I_SCP
  TPCCP_RET_SCP(ordout.c_first, c_first, 17);
  TPCCP_RET_SCP(ordout.c_middle, c_middle, 3);
  TPCCP_RET_SCP(ordout.c_last, c_last, 17);
  TPCCP_RET_SCP(ordout.o_entry_d, entry_date, 20);
}

void do_order_status(cnP, dataP)
   void *cnP;
   orderStatus_data_t *dataP;
{
  struct ordstruct oraStruct;
  int i, rc;

  AUDITLOG(("> do_order_status\n"));

  TPCC_CP(ordin.w_id, w_id);
  TPCC_CP(ordin.d_id, d_id);
  TPCC_CP(ordin.c_id, c_id);
  oraStruct.ordin.bylastname = ((dataP->c_id == 0) ? 1 : 0);
  TPCC_SCP(ordin.c_last, c_last, 17);

  DEBUGP(("Calling TPCord: w_id %d, d_id %d, c_id %d, bylastname %d, c_last %s\n",
oraStruct.ordin.w_id, oraStruct.ordin.d_id, oraStruct.ordin.c_id, oraStruct.ordin.bylastname,
oraStruct.ordin.c_last));

  rc = TPCord(cnP, &oraStruct);
  if ( rc != 0 ) {
     err_printf("Error TPCord: terror %d, rc %d, retry %d, w_id %d, d_id %d, c_id %d, bylastname %d,
c_last %s\n ",
                  oraStruct.ordout.terror, rc, oraStruct.ordout.retry,
                  oraStruct.ordin.w_id, oraStruct.ordin.d_id, oraStruct.ordin.c_id,
                  oraStruct.ordin.bylastname, oraStruct.ordin.c_last);
  }

  copyout_order_status(dataP, &oraStruct);

  dataP->header.returncode = rc == 0 ? TPCC_SUCCESS : oraStruct.ordout.terror;
  AUDITLOG(("< do_order_stats rc %d\n", dataP->header.returncode));
}

void do_stock_level(cnP, dataP)
   void *cnP;
   stockLevel_data_t *dataP;
{
  struct stostruct oraStruct;
  /* What's this comment?? --srs: i only did this one to check the links*/
  int rc;

  AUDITLOG(("> do_stock_level\n"));

  TPCC_CP(stoin.w_id, w_id);
  TPCC_CP(stoin.d_id, d_id);
  TPCC_CP(stoin.threshold, threshold);

  DEBUGP(("Calling TPCsto: w_id %d, d_id %d, threashold %d\n",
                  oraStruct.stoin.w_id, oraStruct.stoin.d_id,
                  oraStruct.stoin.threshold));

  rc = TPCsto(cnP, &oraStruct );
  if ( rc != 0 ) {
     err_printf("Error TPCsto : terror %d, rc %d, retry %d, w_id %d, d_id %d, threashold %d\n",
                  oraStruct.stoout.terror, rc, oraStruct.stoout.retry,
                  oraStruct.stoin.w_id, oraStruct.stoin.d_id,
                  oraStruct.stoin.threshold);
  }

  TPCC_RET_CP(stoout.low_stock, stock_count);

  dataP->header.returncode = rc == 0 ? TPCC_SUCCESS : oraStruct.stoout.terror;

  DEBUGP(("do_stock_lev returning %d\n", dataP->header.returncode));
  AUDITLOG(("< do_stock_level rc %d\n", dataP->header.returncode));

}

void copyin_payment(dataP, oraStructP)
   payment_data_t *dataP;
   struct paystruct *oraStructP;
{
  TPCCP_CP(payin.w_id, w_id);
  TPCCP_CP(payin.d_id, d_id);
  TPCCP_CP(payin.c_w_id, c_w_id);
  TPCCP_CP(payin.c_d_id, c_d_id);
  TPCCP_CP(payin.c_id, c_id);
  oraStructP->payin.bylastname = ((dataP->c_id == 0) ? 1 : 0);
  TPCCP_CP(payin.h_amount, h_amount);
  TPCCP_SCP(payin.c_last, c_last, 17);
}

void copyout_payment(dataP, oraStructP)
   payment_data_t *dataP;
   struct paystruct *oraStructP;
{
  TPCCP_RET_SCP(payout.w_street_1, w_street_1, 21);
  TPCCP_RET_SCP(payout.w_street_2, w_street_2, 21);
  TPCCP_RET_SCP(payout.w_city, w_city, 21);
  TPCCP_RET_SCP(payout.w_state, w_state, 3);
  TPCCP_RET_SCP(payout.w_zip, w_zip, 10);
  TPCCP_RET_SCP(payout.d_street_1, d_street_1, 21);
  TPCCP_RET_SCP(payout.d_street_2, d_street_2, 21);
  TPCCP_RET_SCP(payout.d_city, d_city, 21);
  TPCCP_RET_SCP(payout.d_state, d_state, 3);
  TPCCP_RET_SCP(payout.d_zip, d_zip, 10);
  TPCCP_RET_CP(payout.c_id, c_id);
  TPCCP_RET_SCP(payout.c_first, c_first, 17);
  TPCCP_RET_SCP(payout.c_middle, c_middle, 3);
  TPCCP_RET_SCP(payout.c_last, c_last, 17);
  TPCCP_RET_SCP(payout.c_street_1, c_street_1, 21);
  TPCCP_RET_SCP(payout.c_street_2, c_street_2, 21);
  TPCCP_RET_SCP(payout.c_city, c_city, 21);
  TPCCP_RET_SCP(payout.c_state, c_state, 3);
  TPCCP_RET_SCP(payout.c_zip, c_zip, 10);
  TPCCP_RET_SCP(payout.c_phone, c_phone, 17);
  TPCCP_RET_SCP(payout.c_since, c_date, 11);
  TPCCP_RET_SCP(payout.c_credit, c_credit, 3);
  TPCCP_RET_CP(payout.c_credit_lim, c_credit_lim);
  TPCCP_RET_CP(payout.c_discount, c_discount);
  TPCCP_RET_CP(payout.c_balance, c_balance);
  TPCCP_RET_SCP(payout.c_data, c_data, 201);
  TPCCP_RET_SCP(payout.h_date, pay_date, 20);
  strcpy((char *)dataP->w_name, "W_NAME");
  strcpy((char *)dataP->d_name, "D_NAME");
  /* Ignore c_ytd_payment, c_payment_cnt */
}

void do_payment(cnP, dataP)
   void *cnP;
   payment_data_t *dataP;
{

  struct paystruct oraStruct;
  int firstWh, secondWh;
  int rc;

  AUDITLOG(("> do_payment\n"));

  copyin_payment(dataP, &oraStruct);

#if 0
     err_printf("TPCpay: w_id %d, D_id %d, C_w_id %d, c_id %d, bylastname %d, amount %.2f, c_last %s
(%s)\n",
                  oraStruct.payin.w_id,
                  oraStruct.payin.d_id,
                  oraStruct.payin.c_w_id,
                  oraStruct.payin.c_id,
                  oraStruct.payin.bylastname,
                  oraStruct.payin.h_amount,
                  oraStruct.payin.c_last,
                  dataP->c_last);
#endif


  rc = TPCpay(cnP, &oraStruct);

#if 0
  err_printf("< TPCpay terror %d, rc %d, retry %d\n",
                  oraStruct.payout.terror, rc, oraStruct.payout.retry);
#endif
```

```
  dataP->header.num_rms = 1;
  if (rc != 0 ) {
      err_printf("Error TPCpay: terror %d, rc %d, retry %d, w_id %d, D_id %d, C_w_id %d, c_id %d,
bylastname %d, amount %.2f, c_last %s (%s)\n",
                oraStruct.payout.terror, rc, oraStruct.payout.retry,
                                oraStruct.payin.w_id,
                oraStruct.payin.d_id,
                oraStruct.payin.c_w_id,
                oraStruct.payin.c_id,
                oraStruct.payin.bylastname,
                oraStruct.payin.h_amount,
                oraStruct.payin.c_last ? oraStruct.payin.c_last : "-NULL-",
                                (char *)dataP->c_last ? (char *)dataP->c_last : "-NULL-");
  }

  copyout_payment(dataP, &oraStruct);

  dataP->header.returncode = rc == 0 ? TPCC_SUCCESS : oraStruct.payout.terror;
  AUDITLOG(("< do_payment rc %d\n", dataP->header.returncode));
}

static void copyin_new_order(dataP, oraStructP)
    newOrder_data_t *dataP;
    struct newstruct *oraStructP;
{
    int i;

    TPCCP_CP(newin.w_id, w_id);
    TPCCP_CP(newin.d_id, d_id);
    TPCCP_CP(newin.c_id, c_id);

#define NO_I_CP(ind,a,b) oraStructP->a[ind] = dataP->item[ind].b
#define NO_I_SCP(ind,a,b,len) strncpy((char *)oraStructP->a[ind], (char *)dataP->item[ind].b, len)

    /* tpccpl.c loops over 15 items, we do the same */
    for (i=0; i<15; i++) {
        NO_I_CP(i, newin.ol_i_id, ol_i_id);
        NO_I_CP(i, newin.ol_supply_w_id,ol_supply_w_id);
        NO_I_CP(i, newin.ol_quantity, ol_quantity);
#ifdef DEBUG_SERVER
        fprintf(stderr, "NewOrder: Item %d, supplyWh %d (local %d)\n",
                i,
                oraStructP->newin.ol_supply_w_id[i],
                oraStructP->newin.w_id);
#endif
    }
    /* Ignore all_local field, total_items,
     * tpccpl.c doesnt use them
     */
#undef NO_I_CP
#undef NO_I_SCP
}

void copyout_new_order(dataP, oraStructP)
    newOrder_data_t *dataP;
    struct newstruct *oraStructP;
{
    int i;

    TPCCP_RET_CP(newout.o_id, o_id);
    TPCCP_RET_CP(newout.o_ol_cnt, o_ol_cnt);
    TPCCP_RET_SCP(newout.c_last, c_last, 17);
    TPCCP_RET_SCP(newout.c_credit, c_credit, 3);
    TPCCP_RET_CP(newout.c_discount, c_discount);
    TPCCP_RET_CP(newout.w_tax, w_tax);
    TPCCP_RET_CP(newout.d_tax, d_tax);
    TPCCP_RET_SCP(newout.o_entry_d, entry_date, 20);
    TPCCP_RET_CP(newout.total_amount, total);
    TPCCP_RET_SCP(newout.status, statusline, 26);

#define NO_RET_CP(ind,a,b) dataP->item[ind].b = oraStructP->newout.a[ind]
#define NO_RET_SCP(ind,a,b,len) strncpy((char *)dataP->item[ind].b, (char *)oraStructP->newout.a[ind],
len)

    for (i=0; i<oraStructP->newout.o_ol_cnt && i<15; i++) {
        NO_RET_SCP(i, i_name, name_i, 25);
        NO_RET_CP(i, s_quantity, s_quantity);
                dataP->item[i].brand_generic[0] = oraStructP->newout.brand_generic[i];
                dataP->item[i].brand_generic[1] = '\0';
        NO_RET_CP(i, i_price, price);
        NO_RET_CP(i, ol_amount, ol_amount);
                /* Ignore s_idx and s_dist */
    }
    if (oraStructP->newout.status[0] != '\0') {
                DEBUGP(("TPCnew: status -- %s\n", oraStructP->newout.status));
                dataP->items_valid = 0;
    } else {
                dataP->items_valid = 1;
    }

#undef NO_RET_CP
#undef NO_RET_SCP
}

void do_new_order(cnP, dataP)
  void *cnP;
  newOrder_data_t *dataP;
{
    static int num_calls = 0;
    int i;
    struct newstruct oraStruct;
    int rc;

    AUDITLOG(("> do_new_order\n"));

    /*  Copy the structure into the TPCC structure. */
    copyin_new_order(dataP, &oraStruct);

    DEBUGP(("-> TPCnew %d items to wh %d\n",
                        dataP->o_ol_cnt, dataP->w_id));
    dataP->header.num_rms = 1;

#if 0
        err_printf("Error TPCnew : w_id %d, d_id %d, c_id %d, o_ol_cnt %d (out cnt %d)\n",
            oraStruct.newin.w_id, oraStruct.newin.d_id,
            oraStruct.newin.c_id, dataP->o_ol_cnt, oraStruct.newout.o_ol_cnt);
        for (i=0; i<15; i++) {
            err_printf("ol_i_id %d, ol_supply_w_id %d, ol_quantity %d\n",
                oraStruct.newin.ol_i_id[i], oraStruct.newin.ol_supply_w_id[i],
                oraStruct.newin.ol_quantity[i]);
        }
#endif

    rc = TPCnew(cnP, &oraStruct);

#if 0
    err_printf("< TPCnew terror %d, rc %d, retry %d\n",
            oraStruct.newout.terror, rc, oraStruct.newout.retry);
#endif

    if (rc != 0) {
        err_printf("Error TPCnew : terror %d, rc %d, retry %d, w_id %d, d_id %d, c_id %d, o_ol_cnt %d (out
cnt %d)\n",
                        oraStruct.newout.terror, rc, oraStruct.newout.retry,
                        oraStruct.newin.w_id, oraStruct.newin.d_id,
                        oraStruct.newin.c_id, dataP->o_ol_cnt, oraStruct.newout.o_ol_cnt);
                    for (i=0; i<15; i++) {
                        err_printf("ol_i_id %d, ol_supply_w_id %d, ol_quantity %d\n",
                                oraStruct.newin.ol_i_id[i], oraStruct.newin.ol_supply_w_id[i],
                                oraStruct.newin.ol_quantity[i]);
                    }
    }
    DEBUGP(("-< TPCnew %d\n", rc));

    /* copy out results */
    copyout_new_order(dataP, &oraStruct);

    if (rc == 0) {
                    dataP->header.returncode =
                                dataP->items_valid ? TPCC_SUCCESS : INVALID_NEWO;
#if 0
                    if (dataP->items_valid && (++num_calls % 500) == 0) {
                        int i;
                        err_printf("TPCnew Success: w_id %d, d_id %d, c_id %d, o_ol_cnt %d, Oid %d\n",
                                    oraStruct.newin.w_id, oraStruct.newin.d_id,
                                    oraStruct.newin.c_id, oraStruct.newout.o_ol_cnt,
                    oraStruct.newout.o_id);
                        for (i=0; i<15 && i<oraStruct.newout.o_ol_cnt; i++) {
                                err_printf("   %2d: i_id %i5d, sw_id %4d, qty %d, price %.2f amt %.2f\n",
                                            i, oraStruct.newin.ol_i_id[i],
                                            oraStruct.newin.ol_supply_w_id[i],
                                            oraStruct.newin.ol_quantity[i],
                        oraStruct.newout.i_price[i],
                        oraStruct.newout.ol_amount[i]);
                        }
                    }
#endif
    } else {
                    dataP->header.returncode = oraStruct.newout.terror;
    }

    AUDITLOG(("< do_new_order rc %d\n", dataP->header.returncode));
}
```

client.C

```
/* (C)1997 IBM Corporation */
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <ctype.h>
#include <string.h>
#include <math.h>

#include "screen.h"
#include "encina.h"

extern "C" void set_client_debug_state(void *contextP, int state, int tran);

Encina encina;

extern "C" int client_login(int infd, int outfd, int *w_idP, int *d_idP)
{
```

```
    Thread_data thread(infd, outfd, NULL);
    User_data user_data;
    Login log(&user_data, &thread);
    log.handle();
    *w_idP = user_data.warehouse;
    *d_idP = user_data.district;
    return 0;
}

extern "C" int client_init(int infd, int outfd, int w_id, int d_id,
                                                    void *contextP) {
    int rc = 0;
    Thread_data *threadP = new Thread_data(infd, outfd, contextP);
    Field *menuField = new IntField(threadP, 8);
    User_data user_data;
    Menu  menu(&user_data, threadP);

    user_data.warehouse = w_id;
    user_data.district = d_id;
    menu.present();

    Payment pay(&user_data, threadP);
    Delivery del(&user_data, threadP);
    OrderStatus os(&user_data, threadP);
    StockLevel sl(&user_data, threadP);
    NewOrder no(&user_data, threadP);

    while (rc == 0) {
                int key = menuField->get_key();
                set_client_debug_state(contextP, 1, key - '0');
                switch (key) {
                   case EOF: rc = -1; break;
                   case '1': case 'N': case 'n': rc = no.handle();  break;
                   case '2': case 'P': case 'p': rc = pay.handle(); break;
                   case '3': case 'O': case 'o': rc = os.handle();  break;
                   case '4': case 'D': case 'd': rc = del.handle(); break;
                   case '5': case 'S': case 's': rc = sl.handle();  break;
                   case '\030':
                     position(threadP, 1, 1);
                     threadP->flush(); break;
                   case '9': case 'Q': case 'q': case 'E': case 'e':
                                return(0);
                   default: threadP->write("\a", 1); break;
                }
                set_client_debug_state(contextP, 4, key - '0');
     }
    return 0;
}
```

client.h

```
/*
*               client.h
*
* $Revision: 1.6 $
* $Date: 1998/11/24 21:45:58 $
* $Log:             $
*
*
* $TALog: client.h,v $
* Revision 1.6  1998/11/24  21:45:58  wenjian
* - Add MULTIPLE_INTERFACE compile flag
* [from r1.5 by delta wenjian-23742-TPCC-update-with-Raliegh-code,r1.1]
*
* Revision 1.5  1998/11/09  14:48:13  wenjian
* In an effort to make a new directory structure for TPCC, this delta
* creates two directories: tpcc/client and tpcc/server. All the files
* for this revision are copied from tpcc/sp-tpcc without any change.
* Further change may be needed for some files due to the change of
* the directory structure.
* [added by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.1]
*
* Revision 1.10  1998/10/26  14:41:34  dongfeng
* Add Init command in web client so when something bad happens during
* initialization web client sends back error information and allows
* reinitialization instead of killing IIS server.
*
* Define values of initialization errors.
* [from r1.9 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.3]
*
* Revision 1.9  1998/10/22  20:55:53  wenjian
* [merge of changes from 1.5 to 1.8 into 1.6]
*
* Revision 1.8  1998/10/22  19:18:31  dongfeng
* [from r1.7 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.2]
*
* Revision 1.7  1998/10/08  14:17:59  dongfeng
* Add codes for doing web-based tpcc.
* [from r1.5 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.1]
*
* Revision 1.6  1998/09/26  10:56:25  oz
* - renamed thread_init and thread_done to clnt_thread_init and
*   clnt_thread_done respectively because of name conflicts on AIX 4.3
* [from r1.5 by delta oz-23339-TPCC-update-for-NT, r1.2]
*
* Revision 1.5  1998/01/26  16:19:22  oz
* - moved all the code pertaining to the background
*   thread to its own file and all the data structures
*   to client_utils.h
* [from r1.4 by delta oz-21689-TPCC-move-client-bg-thread-to-separate-file, r1.1]
*
* Revision 1.4  1998/01/23  15:07:43  oz
* - Updated the SP TPCC directory to the latest files used
*   during the SP tpcc audit.
* [from r1.3 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
*
*
*
*/

#ifndef TPCC_CLIENT_H
#define TPCC_CLIENT_H

#if defined (solaris)
#include <dce/pthread.h>
#else   /* solaris */
#include <pthread.h>
#endif  /* solaris */



#ifdef WEB_TPCC_CLIENT
#include <windows.h>

#define MUTEX_T CRITICAL_SECTION
#define MUTEX_LOCK(a) EnterCriticalSection(a)
#define MUTEX_UNLOCK(a) LeaveCriticalSection(a)
#define MUTEX_INIT(mut)   InitializeCriticalSection(mut)
#define MUTEX_DESTROY(mut) DeleteCriticalSection(mut)
#define ERROUT errtpcc

/*initializationstatus */
#define INIT_SUCCESS 0
#define INIT_FAILED  1
#define CELL_NAME_UNAVAILABLE 2
#define MON_RETRIEVEENABLE_FAILED 3
#define MON_INITCLIENT_FAILED 4
#define MON_SECURITYSET_FAILED 5
#define MON_SETREFRESHINTERVAL_FAILED 6
#define NOINFO_TRPC_ERROR 7
#define ENROLL_CLIENT_EXCEPTION 8
#define ERROUT_FILE_NOT_FOUND 9
#define LOG_FILE_NOT_FOUND 10
#define TPCC_KEY_NOT_FOUND 11
#define TERM_ALLOC_FAILED 12

#else
#define MUTEX_T pthread_mutex_t
#define COND_T  pthread_cond_t
#define MUTEX_LOCK(a) pthread_mutex_lock(a)
#define MUTEX_UNLOCK(a) pthread_mutex_unlock(a)
#define COND_WAIT(cond,mut) pthread_cond_wait(cond,mut)
#define COND_SIGNAL(cond) pthread_cond_signal(cond)
#define COND_BROADCAST(cond) pthread_cond_broadcast(cond)
#define MUTEX_INIT(mut)   pthread_mutex_init(mut, pthread_mutexattr_default)
#define COND_INIT(cond)   pthread_cond_init(cond, pthread_condattr_default)
#define MUTEX_DESTROY(mut) pthread_mutex_destroy(mut)
#define COND_DESTROY(cond) pthread_cond_destroy(cond)
#define ERROUT stderr
#endif
/*
 *  Routines and declarations that are common to all clients
 */
void *clnt_thread_init(void);

void clnt_thread_done(void *);

void *terminal_context_init(int);

void send_new_order(void *, newOrder_data_t *);

void send_payment(void *, payment_data_t *);

void send_order_status(void *, orderStatus_data_t  *);

void send_delivery(void*, delivery_data_t *);

void send_stock_level(void *, stockLevel_data_t *);

void send_batch_request(void *contextP, int num, tpcc_data_t *dataP);

void send_unmarshalled(void*contextP,
                                        int tran_type,
                                        int size,
                                        ndr_byte *dataP);

void enroll_client(int id);

#endif  /*  TPCC_CLIENT_H  */
```

client_bg_thread.c

```
/*
*                client_bg_thread.c
*
* $Revision: 1.17 $
* $Date: 1999/05/06 21:28:25 $
* $Log:            $
*
*
* $TALog: client_bg_thread.c,v $
* Revision 1.17  1999/05/06  21:28:25  oz
* - Removed all the .. from the includes
* - Added -I.. to the makefiles instead
* - Moved all the thread related code and connection
*   selection to serverMon.c
* [from r1.13 by delta oz-24309-TPCC-add-oracle8.1-code, r1.5]
*
* Revision 1.13  1999/01/29  20:16:31  wenjian
* Small changes to make check_threads more robust
* [from r1.12 by delta wenjian-23787-TPCC-integrate-code-for-AIX-and-NT, r1.7]
*
* Revision 1.12  1998/12/28  20:07:11  wenjian
* - Change client_info to a pointer pClientInfo for flexibility.
* [from r1.11 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.5]
*
* Revision 1.11  1998/12/14  20:27:53  wenjian
* Made corresponding changes due to data structure change of tran_info_t.
* [from r1.10 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.3]
*
* Revision 1.10  1998/12/11  16:14:18  wenjian
* Add code for checking statistic data in a single variable and collecting
* statistic data based on iStatsFrequency.
*
* - Add new check_threads() in order to use the single statistic var.
* [from r1.9 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.1]
*
* Revision 1.9  1998/12/08  23:03:48  wenjian
* Add (or rename) Makefile for each platform (AIX and NT). Reorganize the
* files a little bit.
*
* - Change path for tran_stat.h
* - Add ifdef before getStatsForm since it is only used by NT
* [from r1.8 by delta wenjian-23787-TPCC-integrate-code-for-AIX-and-NT, r1.1]
*
* Revision 1.8  1998/12/07  20:04:11  wenjian
* Clean up
* [from r1.7 by delta wenjian-23742-TPCC-update-with-Raliegh-code, r1.2]
*
* Revision 1.7  1998/11/24  21:45:58  wenjian
* - Add #ifdef MULTIPLE_INTERFACE
* - Take care special case for check_threads
* [from r1.6 by delta wenjian-23742-TPCC-update-with-Raliegh-code, r1.1]
*
* Revision 1.6  1998/11/09  16:59:35  wenjian
* In this revision, most of the changes are related to the directory of header
* files after directory reorganization. Other changes include adding or removing
* files to put them in the right directories. Makefiles are written for NT
* platform so that nmake is working on NT now. Need a top level Makefile for all
* the directories.
* [from r1.5 by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.2]
*
* Revision 1.5  1998/11/09  14:48:14  wenjian
* In an effort to make a new directory structure for TPCC, this delta
* creates two directories: tpcc/client and tpcc/server. All the files
* for this revision are copied from tpcc/sp-tpcc without any change.
* Further change may be needed for some files due to the change of
* the directory structure.
* [added by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.1]
*
* Revision 1.29  1998/11/06  16:10:54  wenjian
* - Move gettimeofday() in check_threads out of the for loop
* - Minor change for cleanup
* [from r1.28 by delta wenjian-23646-TPCC-clean-up-source-code, r1.1]
*
* Revision 1.28  1998/10/26  15:17:53  dongfeng
* remove #include <sys/times.h> from NT tests
* [from r1.27 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.4]
*
* Revision 1.27  1998/10/22  21:05:37  wenjian
* [merge of changes from 1.12 to 1.26 into 1.25]
*
* Revision 1.26  1998/10/22  19:18:31  dongfeng
* [from r1.24 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.2]
*
* Revision 1.24  1998/10/08  14:17:59  dongfeng
* Add codes for doing web-based tpcc.
* [from r1.12 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.1]
*
* Revision 1.25  1998/10/08  18:03:00  gerstl
* Changes to allow configurations where some servers only service
* specific transaction types. Split transaction interfaces by type.
* [from r1.23 by delta gerstl-23515-TPCC-allow-separate-online-transaction-interfaces, r1.1]
*
* Revision 1.23  1998/09/03  16:07:11  wenjian
* Change GET_USER_SYS_TIME to GET_CLIENT_USAGE
* [from r1.19 by delta wenjian-23183-TPCC-sync-AIX-code-with-Austin, r1.3]
*
* Revision 1.19  1998/08/18  13:35:41  wenjian
* - Clean up including header files
* - Remove client_report() and socket_print_rt_avg()
* - Use #ifdef for the call of getUserSysTime()
* [from r1.16 by delta wenjian-22495-TPCC-add-new-feature-to-monitor-tpcc-clients, r1.7]
*
* Revision 1.16  1998/07/08  18:15:42  wenjian
* Add getUserSysTime().
* [from r1.15 by delta wenjian-22495-TPCC-add-new-feature-to-monitor-tpcc-clients, r1.6]
*
* Revision 1.15  1998/07/02  18:28:51  wenjian
* Change client_status_report to send more information of the
* client process. These changes are matched with changes in
* tpcc_monitor.c.
* [from r1.9 by delta wenjian-22495-TPCC-add-new-feature-to-monitor-tpcc-clients, r1.5]
*
* Revision 1.7  1998/04/29  19:47:40  wenjian
* - Add client_status_report to comunnicate with tpcc_monitor
* - Add socket_print_rt_avg
* [from r1.6 by delta wenjian-22495-TPCC-add-new-feature-to-monitor-tpcc-clients, r1.1]
*
* Revision 1.6  1998/02/17  22:12:40  wenjian
* [merge of changes from 1.3 to 1.4 into 1.5]
*
* Revision 1.4  1998/02/17  16:04:40  oz
* - Split the login into two parts to allow for special logins
* - If the warehouse ID is 0, this is a special login to
*   query the client for status
*
* - check_threads: Return the number of threads
* - New function: client_report
* [from r1.3 by delta oz-21864-TPCC-split-client-login-screen, r1.1]
*
* Revision 1.5  1998/02/17  22:06:59  wenjian
* Add necessary head files for win32
* [from r1.3 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.1]
*
* Revision 1.3  1998/01/29  22:53:34  oz
* - Use pthread delay instead of sleep
* [from r1.2 by delta oz-21749-TPCC-use-pthread-delay-for-bg-thread, r1.1]
*
* Revision 1.2  1998/01/26  20:37:34  oz
* - Remove all the code associated with explicit binding
*
* - Removed include of mon_client_utils.h
* [from r1.1 by delta oz-21697-TPCC-remove-explicit-binding-code, r1.1]
*
* Revision 1.1  1998/01/26  16:19:22  oz
* - moved all the code pertaining to the background
*   thread to its own file and all the data structures
*   to client_utils.h
* [added by delta oz-21689-TPCC-move-client-bg-thread-to-separate-file, r1.1]
*
*
*/

/*
 *  client_bg_thread
 *                A file used for debug purposes only.
 *
 *  It implements a background thread that once a minute checks the
 *  state of all the threads and reports the state of the client.
 *
 */


#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <time.h>
#ifndef WIN32
#include <sys/times.h>  /* for getUserSysTime */
#endif
#if defined (solaris)
#include <dce/pthread.h>
#else
#include <pthread.h>
#endif  /* solaris */
#include <tpm/mon/mon.h>
#include <utils/trace.h>
#include "common/delivery.h"
#ifdef MULTIPLE_INTERFACE
#include "common/neworder.h"
#include "common/payment.h"
#include "common/stocklevel.h"
#include "common/orderstatus.h"
#else
#include "common/tpcc_trans.h"
#endif
#include "common/utilities.h"
#include "client_utils.h"
#include "common/do_tpcc.h"
#include "client.h"
#include "encina_client.h"
#include "tools/tran_stat.h"
#include "common/get_local_time.h"

#ifdef WIN32
#define   read(A,B,C)     recv(A,B,C,0)
#define   write(A,B,C)    send(A,B,C,0)
```

```
#endif

#if 1
#define PRINT_AV(total, num, str)                    \
{                                                    \
  if ((num) > 0) {                                   \
     fprintf(ERROUT, " %s %.0f,", str, (double)(total)/(num));     \
  }                                                  \
}
#else
#define PRINT_AV(a,b,c)
#endif

static void check_threads(total_tran_count_t *tran_ctP, int *numP, int *numInitP);
static struct timeval *client_last_time(thread_descr_t *descrP);
void getUserSysTime(struct timeval *user_time, struct timeval *sys_time);

/*
 *  client_last_time
 *
 *  Each thread maintains the current state it is in and the time
 *  it entered this state.
 *  This routined returns a pointer to the structure in the thread
 *  data that contains the time corresponding to the threads current
 *  state.
 *  Typical use:
 *                  - Set the state, then call gettime on the pointer
 *                    returned by this function.
 */
static struct timeval *client_last_time(thread_descr_t *descrP)
{
   struct timeval *lastTimeP = &descrP->done;
   switch (descrP->state) {
                  case thread_state_init:              /*  Thread is initializing - notrans yet */
                     lastTimeP = &descrP->init;
                     break;
      case thread_state_called:  /*  Tran type was sent by the RTE  */
         lastTimeP = &descrP->called;
         break;
      case thread_state_returned:  /*  Final screen sent to RTE  */
         lastTimeP = &descrP->returned;
         break;
      case thread_state_sent:  /*  Sent to server  */
         lastTimeP = &descrP->sent;
         break;
      case thread_state_received:  /*  Received reply from server */
         lastTimeP = &descrP->received;
         break;
      case thread_state_done:  /*  The thread exited  */
         lastTimeP = &descrP->done;
         break;
      default:
                     err_printf("client_last_time:bad state: %d\n", descrP->state);
         lastTimeP = &descrP->done;
         break;
   }
   return(lastTimeP);
}

void set_client_debug_state(void *contextP, int state, int tran)
{
   thread_info_t *thread_context = (thread_info_t *)contextP;
   struct timezone tz;
   thread_descr_t *descrP = &thread_context->descr;

   descrP->state = state;

   gettimeofday(client_last_time(descrP),&tz);
   if (state == thread_state_called) descrP->tran = tran;
}



/*  How often to report the state of a thread:
 *  If it is in the thread_state_init phase: report if it has been in
 *  that state for more than 5 minutes.
 *  Report if it takes the terminal more thatn 3 minutes to generate the next
 *  transaction.  Otherwise, report if anything takes longer than 60 seconds.
 */
#define THREAD_STATE_REPORT_DELTA(state) \
  ((state) == thread_state_init ? 300 : \
   (state) == thread_state_returned ? 180 : 60)

static char *thread_state_to_str(int state)
{
   char *ret_val = "-Unknown-";
   switch(state) {
                  case thread_state_init:     ret_val = "state_init"; break;
                  case thread_state_called:   ret_val = "state_called"; break;
                  case thread_state_sent:     ret_val = "state_sent"; break;
                  case thread_state_received: ret_val = "state_received"; break;
                  case thread_state_done:     ret_val = "state_done"; break;
                  case thread_state_returned: ret_val = "state_returned"; break;
   }
   return(ret_val);
}

static void print_rt_avg(total_tran_count_t *curP,
                                                total_tran_count_t *prevP,
                                                int type)
{
   int i;
   static char *names[] = {"0", "no", "pa", "os", "dl", "sl"};
   err_printf("%s RT avg: ", type ? "server" : "client");

   for (i=1; i<=MAX_TRAN_TYPE; i++) {
                  int num_trans = curP->tran[i].RTcount - prevP->tran[i].RTcount;
                  double rt_diff = curP->tran[i].RTtotal[type] - prevP->tran[i].RTtotal[type];
                  PRINT_AV(rt_diff, num_trans, names[i]);
   }
   fprintf(ERROUT, "\n");
}


/*
 *  A background thread that keeps tabs on the state of all the
 *  threads of the client.  (For Debug)
 */
static void *bg_thread(void *argP)
{
   static int sleep_time = 60000; /*  in ms  */
   static struct timespec time_wait = {60, 0};
   struct timespec sleep_end;

   total_tran_count_t tran_ct, tran_reported[2];
   int total_newo, total_tran_err;
   struct timeval cur_time;
   struct timezone tz;
   struct timeval time_reported[2];

   gettimeofday(&time_reported[0], &tz);
   time_reported[1] = time_reported[0];

   memset(&tran_reported[0], '\0', 2 * sizeof(tran_reported[0]));

   while (1) {
                  double time_diff1, time_diff2;
                  double tran_diff1, tran_diff2;
      double prev_newo1, prev_newo2;
                  double err_diff1, err_diff2;

                  check_threads(&tran_ct, NULL, NULL);

                  total_tran_err = tran_ct.errors;
      total_newo = tran_ct.tran[NEWO_TRANS].num-tran_ct.tran[NEWO_TRANS].errs;

                  gettimeofday(&cur_time, &tz);
                  time_diff1 = time_diff_ms(&cur_time,&time_reported[0]);
                  prev_newo1 = tran_reported[0].tran[NEWO_TRANS].num -
tran_reported[0].tran[NEWO_TRANS].errs;
                  tran_diff1 = total_newo - prev_newo1;
                  err_diff1 = total_tran_err - tran_reported[0].errors;

                  time_diff2 = time_diff_ms(&cur_time,&time_reported[1]);
                  prev_newo2 = tran_reported[1].tran[NEWO_TRANS].num -
tran_reported[1].tran[NEWO_TRANS].errs;
                  tran_diff2 = total_newo - prev_newo2;
                  err_diff2 = total_tran_err - tran_reported[1].errors;
                  if (total_newo != 0 && tran_diff2 > 0) {
                     err_printf("bg_thread: TPM: %.0f (last %.0f sec), %.0f (last %.0f sec)\n",
                                    tran_diff1 / time_diff1 * 60000, time_diff1 / 1000.,
                                    tran_diff2 / time_diff2 * 60000, time_diff2 / 1000.);
         /* print av server response time for all transactions */
                     print_rt_avg(&tran_ct, &tran_reported[1], 0);
                     print_rt_avg(&tran_ct, &tran_reported[1], 1);
      }

                  if (err_diff2 != 0) {
                     err_printf("bg_thread: errPM %.1f (last %.0f sec)\n",
                                                   err_diff2 / time_diff2 * 60000, time_diff2 / 1000.);
                  }
                  tran_reported[0] = tran_reported[1];
                  tran_reported[1] = tran_ct;
                  time_reported[0] = time_reported[1];
                  time_reported[1] = cur_time;
      pthread_delay_np(&time_wait);
   }
}

#ifdef KEEP_TERMINAL_INFO
static void check_threads(total_tran_count_t *tran_ctP, int *num_threadsP, int *num_threadsInitP)
{
   struct timezone tz;
   int num_per_state[NUM_STATES];
   int total_stuck = 0;
   static int init_printed = 0;
   int total_tran_err;
   int num_active = 0;


   MUTEX_LOCK(&init_lock);

   if (info_list == NULL || (info_list_len < 1)) {
      if (num_threadsP)
                  *num_threadsP = 0;
      if (num_threadsInitP)
                  *num_threadsInitP = 0;
      memset(tran_ctP, '\0', sizeof(*tran_ctP));
```

```
} else {
    int i,j;
    struct timeval cur_time;
            int num_init = 0, num_done = 0;

            for (i=0; i<NUM_STATES; i++) num_per_state[i] = 0;

    gettimeofday(&cur_time, &tz);
    memset(tran_ctP, '\0', sizeof(*tran_ctP));
    for (i=0; i<info_list_len;i++) {
            struct timeval *client_timeP;
            int time_diff;
            thread_descr_t *descrP;
            int delta;
      if (info_list[i] == NULL || !info_list[i]->initialized){
                    continue;
            }
            if (!info_list[i]->done) num_active++;
            descrP = &info_list[i]->descr;
            delta = THREAD_STATE_REPORT_DELTA(descrP->state);
            client_timeP = client_last_time(descrP);

            for (j=1; j<=MAX_TRAN_TYPE; j++) {
          tran_ctP->tran[j].num += info_list[i]->tran[j].num;
          tran_ctP->tran[j].errs += info_list[i]->tran[j].errs;
          tran_ctP->tran[j].RTcount += info_list[i]->tran[j].RTcount;
          tran_ctP->tran[j].RTtotal[0] += info_list[i]->tran[j].RTtotal[0];
          tran_ctP->tran[j].RTtotal[1] += info_list[i]->tran[j].RTtotal[1];
                    tran_ctP->errors += info_list[i]->tran[j].errs;
            }

            time_diff = cur_time.tv_sec - client_timeP->tv_sec;
            DPRINT(("bg_thread: thread %d (index %d) state %s tran %d for %d sec\n",
          info_list[i]->thread_id, i,
          thread_state_to_str(descrP->state),
          descrP->tran,
          time_diff));
            if (descrP->state == thread_state_init) {
                    num_init++;
            } else if (descrP->state == thread_state_done) {
                    num_done++;
            } else if (time_diff > delta) {
        num_per_state[descrP->state] ++;
                    total_stuck++;
                    if (!descrP->printed) {
                        err_printf("bg_thread: thread %d (index %d) state %s tran %d stuck for %d
sec\n",
                info_list[i]->thread_id, i,
                thread_state_to_str(descrP->state),
                descrP->tran,
                time_diff);

                        descrP->printed = 1;
                    }
            } else if (descrP->printed) {
                    err_printf("bg_thread: thread %d (index %d) state %s tran %d unstuck.\n",
                info_list[i]->thread_id, i,
                thread_state_to_str(descrP->state),
                descrP->tran);

                    descrP->printed = 0;
      }
    }
            if (num_threadsP)
              *num_threadsP = num_active;
            if (num_threadsInitP)
              *num_threadsInitP = num_init;

            if (num_init > 0) {
              err_printf("bg_thread: %d threads still in the init state\n",
                                num_init);
            } else if (!init_printed) {
              err_printf("bg_thread: All %d threads are running\n",
                            info_list_len);
              init_printed = 1;
            }
            if (num_active != info_list_len)
              err_printf("%d threads of %d are still active\n",
                            num_active, info_list_len);

            if (num_done > 0) {
              err_printf("bg_thread: %d threads done so far.\n", num_done);
            }
            if (total_stuck > 0) {
              err_printf("bg_thread: Summary %d stuck: ", total_stuck);
              for (i=0; i<NUM_STATES; i++) {
                if (num_per_state[i] > 0) {
                        fprintf(ERROUT, "%d %s, ",
                                num_per_state[i], thread_state_to_str(i));
                }
              }
              fprintf(ERROUT, "\n");
            }
            total_tran_err = 0;
    for (i=0; i<=MAX_TRAN_TYPE; i++)
            total_tran_err += tran_ctP->tran[i].errs;
    if (total_tran_err > 0) {
            err_printf("bg_thread: %d errs: %d no, %d pa, %d os, %d sl\n",
                            total_tran_err,
                            tran_ctP->tran[NEWO_TRANS].errs,
                            tran_ctP->tran[PAYMENT_TRANS].errs,
                            tran_ctP->tran[ORDER_STAT_TRANS].errs,
                            tran_ctP->tran[STOCK_TRANS].errs);
      }
    }
    MUTEX_UNLOCK(&init_lock);
}

#else
static void check_threads(total_tran_count_t *tran_ctP, int *num_threadsP, int *num_threadsInitP)
{
    int i;
    extern total_tran_count_t *pClientInfo;

    if (num_threadsP != NULL)
            *num_threadsP = 0;
    if (num_threadsInitP != NULL)
            *num_threadsInitP = 0;

    memcpy(tran_ctP, pClientInfo, sizeof(total_tran_count_t));

    /* report error info */
    if (pClientInfo->errors > 0) {
      err_printf("bg_thread: %d errs: %d no, %d pa, %d os, %d sl\n",
            pClientInfo->errors,
            pClientInfo->tran[NEWO_TRANS].errs,
            pClientInfo->tran[PAYMENT_TRANS].errs,
            pClientInfo->tran[ORDER_STAT_TRANS].errs,
            pClientInfo->tran[STOCK_TRANS].errs);
    }

}

#endif

void start_bg_debug_thread()
{
    int rc;
    pthread_attr_t attr;
    pthread_t thread;

    if (rc = pthread_attr_create(&attr)) {
      err_printf("start_bg_debug_thread: pthread_attr_create failed: %d\n", rc);
            return;
    }
    if ((rc = pthread_create(&thread,
                attr,
                bg_thread,
                (pthread_addr_t)NULL)) != 0) {
      err_printf("start_bg_debug_thread: pthread_create failed: %d\n", rc);
            return;
    }
    if ( rc = pthread_detach(&thread) != 0) {
            err_printf("start_bd_debug_thread: pthread_detach failed %d\n", rc);
            return;
    }
}



/* client_status_report:
 * mainly copied from bg_thread
 */
void *client_status_report(int fileno)
{
    static struct timespec time_wait = {60, 0};

    total_tran_count_t tran_ct;
    tran_info_t  *curP;
    struct timeval cur_time;
    struct timezone tz;
    char   buf[1024], cmd='\0';
    int    i, cnt=0;

    /* a loop for communication with tpcc_monitor */
    while ( cmd != 'q' ) {
      struct timeval cur_time;
      struct timeval user_time={0,0}, sys_time={0,0};
      struct timezone tz;
            int num_threads, num_threadsInit;

      memset(&tran_ct, 0, sizeof(tran_ct));

        /* read next cmd from the socket */
            read(fileno, buf, 1);
            cmd = buf[0];
            /* DPRINT(("%c\n",cmd)); */
            if (cmd=='q') {
                    break;
            }

            check_threads(&tran_ct, &num_threads, &num_threadsInit);
            gettimeofday(&cur_time, &tz);
#ifdef GET_CLIENT_USAGE
        getUserSysTime(&user_time, &sys_time);
#endif

            /* The tpcc_monitor has to read the data in the same order
```

```
        * to get the correct data.
        */
                prefix_sprintf(buf,"\n");
                write(fileno,buf,strlen(buf));
                sprintf(buf, "%d %d %d %d %d %d\n",
                        cur_time.tv_sec, cur_time.tv_usec,
                        tran_ct.tran[NEWO_TRANS].num-tran_ct.tran[NEWO_TRANS].errs,
                        tran_ct.errors, num_threads, num_threadsInit);
                write(fileno, buf, strlen(buf));
                sprintf(buf, "%d %d %d %d\n", user_time.tv_sec,user_time.tv_usec,
                                sys_time.tv_sec, sys_time.tv_usec);
                write(fileno, buf, strlen(buf));
        for (i=0, curP=tran_ct.tran; i<=MAX_TRAN_TYPE; i++, curP++) {
                        if (i==0) continue;
            sprintf(buf, "%d %d %d %.3f %.3f\n",
               i, curP->RTtotal, curP->errs, curP->RTtotal[0],
               curP->RTtotal[1]);
            write(fileno, buf, strlen(buf));
                }
                write(fileno, ENDMSG, strlen(ENDMSG));

  }
}

/* for AIX only */
void getUserSysTime(struct timeval *user_time, struct timeval *sys_time)
{
#if defined(AIX)
  struct rusage rubuff;

  if (getrusage(RUSAGE_SELF,&rubuff) == 0) {
                  user_time->tv_sec = rubuff.ru_utime.tv_sec;
     user_time->tv_usec = rubuff.ru_utime.tv_usec;

     sys_time->tv_sec = rubuff.ru_stime.tv_sec;
     sys_time->tv_usec = rubuff.ru_stime.tv_usec;

  } else {
     user_time->tv_sec = user_time->tv_usec = 0;
     sys_time->tv_sec = sys_time->tv_usec = 0;
  }
#elif defined(solaris)
  /* WARNING: not test it yet */
  struct tms t;
  static long ticks = 0;
  register long n;

  if (ticks == 0) ticks = sysconf(_SC_CLK_TCK);

  (void)times(&t);

  user_time->tv_sec = t.tms_utime / ticks;
  user_time->tv_usec = (t.tms_utime % ticks) * 1000000 / ticks;

  sys_time->tv_sec = t.tms_stime / ticks;
  sys_time->tv_usec = (t.tms_stime % ticks) * 1000000 / ticks;
#else
  user_time->tv_sec = 0;
  user_time->tv_usec = 1000;

  sys_time->tv_sec = 0;
  sys_time->tv_usec = 1000;
#endif

}

#ifdef WEB_TPCC_CLIENT
extern void getStatsForm(char* outBufP, char* rawStatsP, int interval);

int web_client_status(char* szFormP, int cmd, int interval)
{
  total_tran_count_t tran_ct;
  tran_info_t *curP;
  struct timeval cur_time;
  struct timezone tz;
  int i, cnt=0;
  int num_threads, num_threadsInit;
  char *statusP;
  char tempP[512];

  if(cmd==1)
          statusP = szFormP;
  else
          statusP = tempP;

  *statusP = '\0';
  memset(&tran_ct, 0, sizeof(tran_ct));

  if(cmd == 2) /* quit */
    return 1;

  check_threads(&tran_ct, &num_threads, &num_threadsInit);
  gettimeofday(&cur_time, &tz);

  prefix_sprintf(statusP,"\n");
  sprintf(statusP+strlen(statusP), "%d %d %d %d %d %d\n", cur_time.tv_sec,
        cur_time.tv_usec,
        tran_ct.tran[NEWO_TRANS].num-tran_ct.tran[NEWO_TRANS].errs,
                tran_ct.errors,
                num_threads, num_threadsInit);
  sprintf(statusP+strlen(statusP), "0 0 0 0\n");
  for (i=0, curP=tran_ct.tran; i<=MAX_TRAN_TYPE; i++, curP++) {
    if (i==0) continue;
    sprintf(statusP+strlen(statusP), "%d %d %d %.3f %.3f\n",
            i, curP->RTcount, curP->errs, curP->RTtotal[0],
            curP->RTtotal[1]);
  }

  if(cmd != 1)
          getStatsForm(szFormP, statusP, interval);
  return 0;
}
#endif
```

client_listen.c

```
/*
 *                client_listen.c
 *
 * $Revision: 1.8 $
 * $Date: 1999/05/06 21:28:26 $
 * $Lo:            $
 *
 *
 * $TALog: client_listen.c,v$
 * Revision 1.8  1999/05/06  21:28:26  oz
 * - Removed all the .. from the includes
 * - Added -I.. to the makefiles instead
 * - Moved all the thread related code and connection
 *   selection to serverMon.c
 * [from r1.7 by delta oz-24309-TPCC-add-oracle8.1-code, r1.5]
 *
 * Revision 1.7  1999/01/29  20:16:32  wenjian
 * Change logprintf to err_printf
 * [from r1.6 by delta wenjian-23787-TPCC-integrate-code-for-AIX-and-NT, r1.7]
 *
 * Revision 1.6  1998/11/09  16:59:36  wenjian
 * In this revision, most of the changes are related to the directory of header
 * files after directory reorganization. Other changes include adding or removing
 * files to put them in the right directories. Makefiles are written for NT
 * platform so that nmake is working on NT now. Need a top level Makefile for all
 * the directories.
 * [from r1.5 by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.2]
 *
 * Revision 1.5  1998/11/09  14:48:14  wenjian
 * In an effort to make a new directory structure for TPCC, this delta
 * creates two directories: tpcc/client and tpcc/server. All the files
 * for this revision are copied from tpcc/sp-tpcc without any change.
 * Further change may be needed for some files due to the change of
 * the directory structure.
 * [added by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.1]
 *
 * Revision 1.30  1998/09/26  10:56:25  oz
 * - renamed thread_init and thread_done to clnt_thread_init and
 *   clnt_thread_done respectively because of name conflicts on AIX4.3
 * [from r1.22 by delta oz-23339-TPCC-update-for-NT, r1.2]
 *
 * Revision 1.22  1998/08/18  14:38:38  wenjian
 * Minor change
 * [from r1.18 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.4]
 *
 * Revision 1.18  1998/04/29  19:47:41  wenjian
 * - Use fd instead of stream on NT
 * - Add code to consider tpcc_monitor as a special client login
 * - Use TRY and CATCH_ALL to deal with exceptions
 * [from r1.17 by delta wenjian-22495-TPCC-add-new-feature-to-monitor-tpcc-clients, r1.1]
 *
 * Revision 1.17  1998/02/17  22:13:28  wenjian
 * [merge of changes from 1.14 to 1.15 into 1.16]
 *
 * Revision 1.15  1998/02/17  16:04:41  oz
 * - Split the login into two parts to allow for special logins
 * - If the warehouse ID is 0, this is a special login to
 *   query the client for status
 *
 * - First, login
 *   If the w_id is bigger than 0: normal thread.
 *   Otherwise, call client_report.
 * [from r1.14 by delta oz-21864-TPCC-split-client-login-screen, r1.1]
 *
 * Revision 1.16  1998/02/17  22:06:59  wenjian
 * Add head files and define macros for win32
 * [from r1.14 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.1]
 *
 * Revision 1.14  1998/01/28  22:24:48  oz
 * [from r1.13 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.4]
 *
 * Revision 1.13  1998/01/26  16:19:22  oz
 * - moved all the code pertaining to the background
 *   thread to its own file and all the data structures
 *   to client_utils.h
 * [from r1.12 by delta oz-21689-TPCC-move-client-bg-thread-to-separate-file, r1.1]
 *
 * Revision 1.12  1998/01/24  14:17:04  oz
 * - User server name to identify server and name delivery file
 * - Use env variable HOME instead of /home/encina if HOME is set
 *
```

```
* - Print the thread ID on thread exit as well
* [from r1.11 by delta oz-21687-TPCC-use-server-name-to-identify-process, r1.1]
*
* Revision 1.11  1998/01/23  15:07:44  oz
* - Updated the SP TPCC directory to the latest files used
*   during the SP tpcc audit.
* [from r1.10 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
*
*
*  Exported functions:
*                   make_connections
*
*  Private functions:
*                   process_terminal
*
*
*/

/*   client_listen.c
*                   Code in the client that listens for requests from the
*                   terminal processes and submits them for processing.
*
*                   There is one listening function:make_connection.
*                   That function calls cnm_ManageConnection which never returns
*                   and so it is best to call it in its own independent thread.
*
*                   As soon as cnm_ManageConnections receives a connection it
*  starts a new thread and calls process_terminal in that
*  thread passing in the file descriptor for the new connection.
*
*                   Note that the client does not need to know in advance how many
*                   terminals it will talk to.
*
*                   The function process_terminal reads initializes the thread
*  and then calls client_init to process all the requests from
*  that terminal.
*/


#include <stdlib.h>
#ifdef WIN32
#include <io.h>
#endif
#include <stdio.h>
#include <sys/types.h>
#include <tc/tc.h>
#include "common/do_tpcc.h"
#include "common/tpcc_type.h"

#if defined (solaris)
#include <dce/pthread.h>
#else   /* solaris */
#include <pthread.h>
#endif  /* solaris */

#include "client_utils.h"
#ifdef WIN32
#include "cnm.h"
#else
#include <cnm/cnm.h>
#endif

#ifdef WIN32
#define         close           _close
#define         fileno          _fileno
#endif
/*
 *  State about the terminal stored by the terminal thread
 *  work_entry: The work entry to be used by this terminal thread.
 */
typedef struct  {
    int profiling;
    int terminal_id;
    void *handle_contextP;
} terminal_context_t;

/**
 **  Function Prototypes
 **/
static void process_terminal(cnm_arg_t *argP);

extern void client_init(int, int, int, int, void *);
extern void client_login(int, int, int *, int *);

/*
 *  process_terminal
 *
 *                  The argument we get is a file descriptor for a terminal
 *                  process.  We read from that file to receive input and send
 *                  output back to that file.
 */
static void process_terminal(cnm_arg_t *argP)
{
    int w_id, d_id;
    terminal_context_t terminal_context;
    tpcc_data_t tran_data;
    int fdIn;
    pthread_t thread = pthread_self();
    int thread_id = pthread_getunique_np(&thread);
    struct timespec rand_sleep;
#if defined(_AIX)
    tid_t tid = thread_self();
#else
    int tid = thread_id;
#endif

#ifdef WIN32
    fdIn = argP->fd;
#else
    fdIn = fileno(argP->stream);
#endif /* WIN32 */

    /*
     *  Default terminal context
     *  This may be updated later by the terminal
     */
    terminal_context.terminal_id= -1;
    terminal_context.profiling= 0;

    TRY {
        client_login(fdIn, fdIn, &w_id, &d_id);
        if (w_id > 0) {
            /*  Initialize the server handle and other thread structures  */
            terminal_context.handle_contextP = (void *)clnt_thread_init();

            err_printf("Tid: %d (0x%x) w_id %d, d_id %d\n", tid, tid, w_id, d_id);
                    client_init(fdIn, fdIn, w_id, d_id, terminal_context.handle_contextP);
                    err_printf("Thread done - Tid %d (0x%x)\n", tid, tid);
                    clnt_thread_done(terminal_context.handle_contextP);
        } else {
                    err_printf("Starting Auxiliary Thread, Tid %d (0x%x)\n", tid, tid);
                    client_status_report(fdIn);
                    err_printf("End of Auxiliary Thread, Tid %d (0x%x)\n", tid, tid);
        }
    } CATCH_ALL {
                err_printf("An exception happened\n");
                logprintf("End of Auxiliary Thread, Tid %d (0x%x)\n", tid, tid);
    }
    ENDTRY

    close(fdIn);
}

/*
 *  make_connections
 *
 *    Listen for connections on a socket.
 *    Whenever a connection is made, start a thread to talk
 *    to the terminal.
 *                  This functions is spawned on its own thread.
 */
void make_connections(argP)
  void *argP;
{
    int port = (int)argP;
    char port_descr[28];
    int rc;

    DPRINT(("Using socket %d\n", port));
    err_printf("Using thread stack size default\n");
    sprintf(port_descr, "ncacn_ip_tcp[%d]", port);
    rc = cnm_ManageConnections(port_descr,
                                                    (cnm_userRountine_t)process_terminal,
                                                    NULL,
                                                    0,          /*  Max Connections  */
                                                    1);   /*  Spawn threads   */
    err_printf("cnm_ManageConnections returned %d\n", rc);
}
```

client_listen.h

```
/*
 *                  client_listen.h
 *
 * $Revision: 1.5 $
 * $Date: 1998/11/09 14:48:14 $
 * $Log:            $
 *
 *
 * $TALog: client_listen.h,v$
 * Revision 1.5  1998/11/09  14:48:14  wenjian
 * In an effort to make a new directory structure for TPCC, this delta
 * creates two directories: tpcc/client and tpcc/server. All the files
 * for this revision are copied from tpcc/sp-tpcc without any change.
 * Further change may be needed for some files due to the change of
 * the directory structure.
 * [added by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.1]
 *
 * Revision 1.1  1997/04/20  11:57:55  oz
 * - This is the code base modified at IBM Poughkeepsie
 *   by Ofer Zajicek and Radha Sivaramakrishnan for the
 *   SP scaling test for TPCC.
 * [added by delta oz-19782-TPCC-add-ibm-sp-code, r1.1]
 *
 * Revision 1.1  1995/07/09  18:12:10  oz
 * - Modified the client side of the TPCC benchmark to have multithreaded
 *   clients.  There is a terminal process for each terminal -- when
 *   not using the terminal emulator each terminal process emulates one
```

```
*   terminal.  The terminal processes communication with the client
*   process using a unix socket.
*
*   On the client side there is a thread for each terminal process.
*   That thread receives the request from the terminal and puts it on
*   a queue.  There is one processing thread that dequeues the requests
*   and sends them to the server for processing.
* [added by delta oz-15875-TPCC-reduce-the-number-of-clients, r1.1]
*
*
*
*/

/*   client_listen.h
*/

#ifndef TPCC_CLIENT_LISTEN_H
#define TPCC_CLIENT_LISTEN_H


void make_connections(void *argP);

#endif /* TPCC_CLIENT_LISTEN_H */
```

client_main.c

```
#include "string.h"
#include "tpcc.h"

extern void client_init(int infd, int outfd, int w_id, int d_id, void *conP);
extern void client_login(int infd, int outfd, int *w_idP, int *d_idP);

main()
{
    int w_id, d_id;
    client_login(0, 1, &w_id, &d_id);
    client_init(0, 1, w_id, d_id, (void *)0);
}


int send_new_order(void *contextP, NewOrder_data *data) {
    int             i;

    data->s_W_ID = 11;
    data->s_D_ID = 22;
    data->s_C_ID = 3333;
    strcpy((char *)data->s_C_LAST, "1234567890123456");
    strcpy((char *)data->s_C_CREDIT, "BC");
    data->s_C_DISCOUNT = 0.1556;
    data->s_O_ID = 4444;
    strcpy((char *)data->s_O_ENTRY_D, "1992-10-2 12:33:11");
    strcpy((char *)data->s_status_line, "123456789012345678901234");
    data->s_total_amount = 12.98;
    data->s_transtatus = 0;
    data->s_W_TAX = 0.1234;
    data->s_D_TAX = 0.5678;

    for (i=0; i < data->s_O_OL_CNT; i++) {
                strcpy((char *)data->item[i].s_I_NAME, "123456789012345678901234");
                data->item[i].s_S_QUANTITY = i + 1;
                data->item[i].s_brand_generic[0] = 'B';
                data->item[i].s_I_PRICE = i + 1;
                data->item[i].s_OL_AMOUNT = i + 1;
    }
    return 0;
}

int send_payment(void *contextP, Payment_data *data) {
    data->s_W_ID = 11;
    data->s_D_ID = 22;
    data->s_C_ID = 3333;
    data->s_C_W_ID = 44;
    data->s_C_D_ID = 55;
    data->s_H_AMOUNT = 9.55;
    strcpy((char *)data->s_W_STREET_1, "1234567890123456789O");
    strcpy((char *)data->s_W_STREET_2, "1234567890123456789O");
    strcpy((char *)data->s_W_CITY, "123456789012345678901");
    strcpy((char *)data->s_W_STATE, "PR");
    strcpy((char *)data->s_W_ZIP, "123456789");
    strcpy((char *)data->s_D_STREET_1, "1234567890123456789O");
    strcpy((char *)data->s_D_STREET_2, "1234567890123456789O");
    strcpy((char *)data->s_D_CITY, "123456789012345678901");
    strcpy((char *)data->s_D_STATE, "PR");
    strcpy((char *)data->s_D_ZIP, "123456789");
    strcpy((char *)data->s_C_FIRST, "1234567890123456");
    strcpy((char *)data->s_C_MIDDLE, "12");
    strcpy((char *)data->s_C_LAST, "1234567890123456");
    strcpy((char *)data->s_C_STREET_1, "1234567890123456789O");
    strcpy((char *)data->s_C_STREET_2, "1234567890123456789O");
    strcpy((char *)data->s_C_CITY, "123456789012345678901");
    strcpy((char *)data->s_C_STATE, "PR");
    strcpy((char *)data->s_C_ZIP, "123456789");
    strcpy((char *)data->s_C_PHONE, "1234567890123456");
    strcpy((char *)data->s_C_SINCE, "1992-23-22 21:11:11");
    strcpy((char *)data->s_H_DATE, "1992-10-2 12:33:11");
    strcpy((char *)data->s_C_CREDIT, "BC");
    data->s_C_CREDIT_LIM = 5000;
    data->s_C_DISCOUNT = 0.10;
    data->s_C_BALANCE = 122.10;
    strcpy((char *)data->s_C_DATA,
"1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890");
    return 0;
}

int send_order_status(void *contextP, OrderStatus_data *data) {
    int             i;

    data->s_W_ID = 11;
    data->s_D_ID = 22;
    data->s_C_ID = 3333;
    strcpy((char *)data->s_C_FIRST, "1234567890123456");
    strcpy((char *)data->s_C_MIDDLE, "12");
    strcpy((char *)data->s_C_LAST, "1234567890123456");
    data->s_C_BALANCE = 122.10;
    data->s_O_ID = 44;
    strcpy((char *)data->s_O_ENTRY_D,"1992-10-2 12:33:11");
    data->s_O_CARRIER_ID = 55;
    data->s_ol_cnt = 10;

    for (i=0; i < data->s_ol_cnt; i++) {
                data->item[i].s_OL_SUPPLY_W_ID = i + 1;
                data->item[i].s_OL_I_ID = i + 1;
                data->item[i].s_OL_QUANTITY = i + 1;
                data->item[i].s_OL_AMOUNT = i + 1;
                strcpy((char *)data->item[i].s_OL_DELIVERY_D, "1992-10-2 12:33:11");
    }
    return 0;
}

int send_delivery(void *contextP, Delivery_data *data) {
    strcpy((char *)data->s_exec_status, "Delivery has been queued");
    return 0;
}

int send_stock_level(void *contextP, StockLevel_data *data) {
    data->s_low_stock = 22;
    return 0;
}
```

client_utils.c

```
/*
 *
 *              client_utils.c
 *
 * $Revision: 1.9 $
 * $Date: 1999/05/06 21:28:26 $
 * $Log:        $
 *
 *
 *
 *
 * $TALog: client_utils.c,v$
 * Revision 1.9  1999/05/06  21:28:26  oz
 * - Removed all the .. from the includes
 * - Added -I.. to the  makefiles instead
 * - Moved all the thread related code and connection
 *   selection to serverMon.c
 * [from r1.7 by delta oz-24309-TPCC-add-oracle8.1-code, r1.5]
 *
 * Revision 1.7  1998/12/11  16:37:57  wenjian
 * Move some common functions from client/client_utils.c to common/tpcc_utils.c.
 * In this version, we only move time_diff_ms(). Need some work in order to
 * move other functions like ERROUT.
 *
 * - Move time_diff_ms() to common/tpcc_utils.c
 * [from r1.6 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.2]
 *
 * Revision 1.6  1998/11/09  16:59:36  wenjian
 * In this revision, most of the changes are related to the directory of header
 * files after directory reorganization. Other changes include adding or removing
 * files to put them in the right directories. Makefiles are written for NT
 * platform so that nmake is working on NT now. Need a top level Makefile for all
 * the directories.
 * [from r1.5 by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.2]
 *
 * Revision 1.5  1998/11/09  14:48:14  wenjian
 * In an effort to make a new directory structure for TPCC, this delta
 * creates two directories: tpcc/client and tpcc/server. All the files
 * for this revision are copied from tpcc/sp-tpcc without any change.
 * Further change may be needed for some files due to the change of
 * the directory structure.
 * [added by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.1]
 *
 * Revision 1.9  1998/10/08  14:18:00  dongfeng
 * Add codes for doing web-based tpcc.
 * [from r1.7 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.1]
 *
 * Revision 1.7  1998/04/29  19:47:42  wenjian
 * - Add prefix_sprintf
 * - Remove ENCINA_C_CALLING_CONVENTION from err_printf
 * [from r1.6 by delta wenjian-22495-TPCC-add-new-feature-to-monitor-tpcc-clients, r1.1]
 *
 * Revision 1.6  1998/02/17  22:07:00  wenjian
 * Minor changes for NT
```

```
* [from r1.5 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.1]
*
* Revision 1.5  1998/01/24  14:17:04  oz
* - User server name to identify server and name delivery file
* - Use env variable HOME instead of /home/encina if HOME is set
*
* - Flush the logfile after each write
* [from r1.4 by delta oz-21687-TPCC-use-server-name-to-identify-process, r1.1]
*
* Revision 1.4  1998/01/23  15:07:46  oz
* - Updated the SP TPCC directory to the latest files used
*   during the SP tpcc audit.
* [from r1.3 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
*
*
*    client_utils.c
*                Generic utilities used by the client processes
*/

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

#if defined (solaris)
#include <dce/pthread.h>
#else   /* solaris */
#include <pthread.h>
#endif

#include "common/databuf.h"
#include "client_utils.h"
#include "common/do_tpcc.h"
#include "common/tpcc_type.h"


#define CASE(a) case a: retVal = #a; break

int print_thread_id = 1;
extern int user_id;
extern char *user_code;
/*
 *  Translate the tpcc return code to a string value
 */
static char *TpccRcToStr(rc)
  tpcc_rc_t rc;
{
   char *retVal;
   switch (rc) {
                CASE(INVALID_NEWO);
                CASE(INVALID_HANDLE);
                CASE(SQL_ERROR);
                CASE(TRPC_ERROR);
                CASE(DCE_ERROR);
                CASE(NO_SUCH_LAST_NAME);
                CASE(INVALID_TRAN_TYPE);
                CASE(TPCC_ERROR_BEGIN_NEWO);
                CASE(TPCC_ERROR_DECL_NEWO_SEL_ITEM);
                CASE(TPCC_ERROR_OPEN_NEWO_SEL_ITEM);
                CASE(TPCC_ERROR_OPEN_DIST_NEWO_SEL_ITEM);
                CASE(TPCC_ERROR_FETCH_NEWO_SEL_ITEM);
                CASE(TPCC_ERROR_FETCH_DIST_NEWO_SEL_ITEM);
                CASE(TPCC_ERROR_PREP_NEWO_SEL_STCK);
                CASE(TPCC_ERROR_DECL_NEWO_SEL_STCK);
                CASE(TPCC_ERROR_OPEN_NEWO_SEL_STCK);
                CASE(TPCC_ERROR_OPEN_DIST_NEWO_SEL_STCK);
                CASE(TPCC_ERROR_FETCH_NEWO_SEL_STCK);
                CASE(TPCC_ERROR_FETCH_DIST_NEWO_SEL_STCK);
                CASE(TPCC_ERROR_NEWO_SELECT);
                CASE(TPCC_ERROR_NEWO_UPD_STCK);
                CASE(TPCC_ERROR_DIST_NEWO_UPD_STCK);
                CASE(TPCC_ERROR_NEWO_SELECT_2);
                CASE(TPCC_ERROR_DECL_NEWO_SEL_CUST);
                CASE(TPCC_ERROR_OPEN_NEWO_SEL_CUST);
                CASE(TPCC_ERROR_OPEN_DIST_NEWO_SEL_CUST);
                CASE(TPCC_ERROR_FETCH_NEWO_SEL_CUST);
                CASE(TPCC_ERROR_FETCH_DIST_NEWO_SEL_CUST);
                CASE(TPCC_ERROR_DECL_NEWO_SEL_DIST);
                CASE(TPCC_ERROR_OPEN_NEWO_SEL_DIST);
                CASE(TPCC_ERROR_OPEN_DIST_NEWO_SEL_DIST);
                CASE(TPCC_ERROR_FETCH_NEWO_SEL_DIST);
                CASE(TPCC_ERROR_FETCH_DIST_NEWO_SEL_DIST);
                CASE(TPCC_ERROR_PREP_NEWO_INS_OL);
                CASE(TPCC_ERROR_DECL_NEWO_INS_OL);
                CASE(TPCC_ERROR_OPEN_NEWO_INS_OL);
                CASE(TPCC_ERROR_OPEN_DIST_NEWO_INS_OL);
                CASE(TPCC_ERROR_PUT_NEWO_INS_OL);
                CASE(TPCC_ERROR_PUT_DIST_NEWO_INS_OL);
                CASE(TPCC_ERROR_DECL_NEWO_SEL_WARE);
                CASE(TPCC_ERROR_OPEN_NEWO_SEL_WARE);
                CASE(TPCC_ERROR_OPEN_DIST_NEWO_SEL_WARE);
                CASE(TPCC_ERROR_FETCH_NEWO_SEL_WARE);
                CASE(TPCC_ERROR_FETCH_DIST_NEWO_SEL_WARE);
                CASE(TPCC_ERROR_EXECUTE_NEWO_UPD_INS);
                CASE(TPCC_ERROR_UPDATE_NEWO_NEXT_OID);
                CASE(TPCC_ERROR_PREP_NEWO_INS);
                CASE(TPCC_ERROR_EXECUTE_DIST_NEWO_INS);
                CASE(TPCC_ERROR_EXECUTE_NEWO_COMMIT);
                CASE(TPCC_ERROR_ROLLBACK_NEWO);
                CASE(TPCC_ERROR_REMOTE_OL_SELECT);
                CASE(TPCC_ERROR_REMOTE_OL_UPDATE);
                CASE(TPCC_ERROR_OPEN_ORDS_CNT_CID);
                CASE(TPCC_ERROR_FETCH_ORDS_CNT_CID);
                CASE(TPCC_ERROR_OPEN_ORDS_SEL_CLAST);
                CASE(TPCC_ERROR_FETCH_ORDS_SEL_CLAST);
                CASE(TPCC_ERROR_OPEN_ORDS_SEL_CID);
                CASE(TPCC_ERROR_FETCH_ORDS_SEL_CID);
                CASE(TPCC_ERROR_OPEN_ORDS_SEL_OLDORD);
                CASE(TPCC_ERROR_FETCH_ORDS_OLDORD);
                CASE(TPCC_ERROR_OPEN_ORDS_SEL_OL);
                CASE(TPCC_ERROR_FETCH_ORDS_SEL_OL);
                CASE(TPCC_ERROR_EXECUTE_ORDS_COMMIT);
                CASE(TPCC_ERROR_OPEN_DELIVERY_OLDEST_OID);
                CASE(TPCC_ERROR_FETCH_DELIVERY_OLDEST_OID);
                CASE(TPCC_ERROR_EXECUTE_DELIVERY_COMMIT);
                CASE(TPCC_ERROR_OPEN_DELIVERY_SEL_ORD);
                CASE(TPCC_ERROR_FETCH_DELIVERY_SEL_ORD);
                CASE(TPCC_ERROR_OPEN_DELIVERY_SEL_SUM_OL);
                CASE(TPCC_ERROR_FETCH_DELIVERY_SEL_SUM_OL);
                CASE(TPCC_ERROR_EXECUTE_DELIVERY_EXEC_DVRY);
                CASE(TPCC_ERROR_SELECT_DELIVERY_ORDER_ID);
                CASE(TPCC_ERROR_SELECT_DELIVERY_CARRIER_ID);
                CASE(TPCC_ERROR_SELECT_DELIVERY_BALANCE);
                CASE(TPCC_ERROR_OPEN_STOCKLEVEL_SEL_OID);
                CASE(TPCC_ERROR_FETCH_STOCKLEVEL_SEL_OID);
                CASE(TPCC_ERROR_OPEN_STOCKLEVEL_CNT_SID);
                CASE(TPCC_ERROR_FETCH_STOCKLEVEL_CNT_SID);
                CASE(TPCC_ERROR_OPEN_STOCKLEVEL_FIND);
                CASE(TPCC_ERROR_FETCH_STOCKLEVEL_FIND);
                CASE(TPCC_ERROR_EXECUTE_STOCKLEVEL_COMMIT);
                CASE(TPCC_ERROR_OPEN_PAYMENT_CNT_CID);
                CASE(TPCC_ERROR_FETCH_PAYMENT_CNT_CID);
                CASE(TPCC_ERROR_OPEN_PAYMENT_SEL_CLAST);
                CASE(TPCC_ERROR_FETCH_PAYMENT_SEL_CLAST);
                CASE(TPCC_ERROR_OPEN_PAYMENT_SEL_CID);
                CASE(TPCC_ERROR_FETCH_PAYMENT_SEL_CID);
                CASE(TPCC_ERROR_DECL_PAYMENT_SEL_DIST);
                CASE(TPCC_ERROR_OPEN_PAYMENT_SEL_DIST);
                CASE(TPCC_ERROR_OPEN_DIST_PAYMENT_SEL_DIST);
                CASE(TPCC_ERROR_FETCH_PAYMENT_SEL_DIST);
                CASE(TPCC_ERROR_FETCH_DIST_PAYMENT_SEL_DIST);
                CASE(TPCC_ERROR_DECL_PAYMENT_SEL_WARE);
                CASE(TPCC_ERROR_OPEN_PAYMENT_SEL_WARE);
                CASE(TPCC_ERROR_OPEN_DIST_PAYMENT_SEL_WARE);
                CASE(TPCC_ERROR_FETCH_PAYMENT_SEL_WARE);
                CASE(TPCC_ERROR_FETCH_DIST_PAYMENT_SEL_WARE);
                CASE(TPCC_ERROR_EXECUTE_PAYMENT_UPD_CUST_LAST);
                CASE(TPCC_ERROR_EXECUTE_PAYMENT_UPD_CUST_ID);
                CASE(TPCC_ERROR_COMMIT_PAYMENT_UPD_CUST);
                CASE(TPCC_ERROR_SELECT_PAYMENT_W_YTD);
                CASE(TPCC_ERROR_SELECT_PAYMENT_D_YTD);
                CASE(TPCC_ERROR_BEGIN_PAYMENT);
                CASE(TPCC_ERROR_EXECUTE_PAYMENT_COMMIT);
     default: retVal = "-Unknown-"; break;
   }
   return(retVal);
}

/*
 *  get_thread_id
 *      A function that returns the thread ID of the current thread
 */
int get_thread_id()
{
#ifdef WEB_TPCC_CLIENT
   return(GetCurrentThreadId());
#else
   pthread_t thread = pthread_self();
   int thread_id = pthread_getunique_np(&thread);
   return(thread_id);
#endif
}

#define A_CASE(a,b)  case a: retVal = b; break
/*
 *  Translate the transaction code to its name - for formatting
 */
char *clientUtils_TranCodeToName(type)
  int type;
{
   char *retVal = "-Unknown-";
   switch (type) {
                A_CASE(NEWO_TRANS,      "NEWOR");
                A_CASE(PAYMENT_TRANS,   "PAYMN");
                A_CASE(ORDER_STAT_TRANS, "ORDER");
                A_CASE(DELIVERY_TRANS,  "DELIV");
                A_CASE(STOCK_TRANS,     "STOCK");
   }
   return(retVal);
}

/*
 *  Print the return status of a TPCC transaction
 *  and the corresponding SQL codes and ISAM codes
 */
void clientUtils_ReportReturn(msg, statusP)
  char *msg;
  data_header *statusP;
```

```
{
  switch (statusP->returncode) {
    case SUCCESS_CODE:
                err_printf( "After %s, rc = %d\n", msg, statusP->returncode);
                break;
    case SQL_ERROR:
                err_printf("ERROR: After %s, rc = SQL_ERROR, SQL=%d, ISAM=%d\n", msg,
                                statusP->sql_code,
                                statusP->isam_code);
                break;
    case INVALID_NEWO:
                err_printf("After %s, rc = INVALID_NEWO\n", msg);
                break;
    case DCE_ERROR:
                err_printf("ERROR: After %s, rc = DCE_ERROR\n", msg);
                break;
    case TRPC_ERROR:
                err_printf("ERROR: After %s, rc = TRPC_ERROR\n", msg);
                break;
    case NO_SUCH_LAST_NAME:
                err_printf("After %s, rc = NO_SUCH_LAST_NAME.\n", msg);
                break;
    case DISTRIBUTED_TRAN_FAILED:
                err_printf("After %s, rc = DISTRIBUTED_TRAN_FAILED.\n", msg);
                break;
    default:
                err_printf("ERROR: After %s, rc = %s (%d), SQL=%d, ISAM=%d\n", msg,
                                TpccRcToStr(statusP->returncode),
                                statusP->returncode,
                                statusP->sql_code, statusP->isam_code);
                break;
  }
}


/*
*  clientUtils_SetReturnCode
*
*  Set the return code in the dataP union.
*  dataP is a pointer to a union of all the transaction types.
*  Each member of the union has a header field that contains
*  a return code.  Set the  returncode value of the header field
*  for dataP to be code.
*/
void clientUtils_SetReturnCode(dataP, code)
    tpcc_data_t *dataP;
    tpcc_rc_t code;
{
  switch (dataP->tran_type) {
    case NEWO_TRANS: {
                  newOrder_data_t *ptr = &dataP->data.new_order;
                  ptr->header.returncode = code;
                  break;
    }
    case PAYMENT_TRANS: {
                  payment_data_t *ptr = &dataP->data.payment;
                  ptr->header.returncode = code;
                  break;
    }
    case ORDER_STAT_TRANS: {
                  orderStatus_data_t  *ptr = &dataP->data.order_status;
                  ptr->header.returncode = code;
                  break;
    }
    case DELIVERY_TRANS: {
                  delivery_data_t *ptr = &dataP->data.delivery;
                  ptr->header.returncode = code;
                  break;
    }
    case STOCK_TRANS: {
                  stockLevel_data_t *ptr = &dataP->data.stock_level;
                  ptr->header.returncode = code;
                  break;
    }
  }
}

/*
*  get_prefix
*                Format the output prefix for printing:
*                It contains the user_id, 'C' or 'T' depending on whether it
*                is a terminal or a client and optional a thread identifier
*                The prefix is written in the buffer passed in by the caller.
*/
void get_prefix(buffer)
 char *buffer;
{
  if (print_thread_id) {
                int thread_id = get_thread_id();
                sprintf(buffer,  "%s(%d-%s-%d)%s",
                                user_id < 10 ? "  " : user_id < 100 ? " " : "",
                                user_id,
                                user_code,
                                thread_id,
                                thread_id < 10 ? " " : "");
  } else {
                sprintf(buffer,  "%s(%2d-%s)",
                                user_id < 10 ? " " : "", user_id, user_code);
  }
}

/*
 * err_printf
 *      A var-arg function that appends the current time and
 *      other data to the print request and sends it to  stderr
 *      if it is not a web client, to a file if it is
 */
void err_printf(char *format, ...)
{
    time_t cur_timet;
    char time_str[30];
    char line_prefix[50];
    va_list ap;

    va_start(ap, format);

    cur_timet = time(&cur_timet);
    strftime(time_str, 29, "%X", localtime(&cur_timet));

    get_prefix(line_prefix);

    fprintf(ERROUT, "%s %s - ", line_prefix,time_str);
    vfprintf(ERROUT, format, ap);
#ifdef WEB_TPCC_CLIENT
    fflush(ERROUT);
#endif

    va_end(ap);
}

/*
 * logprintf
 *      A var-arg function that prints both to standard error to  to
 *      the log file.  It prepends every line with the current time
 *      and the user id.
 */
void logprintf( char *format, ...)
{
    time_t cur_timet;
    char time_str[30];
    char line_prefix[50];
    va_list ap;

    va_start(ap, format);

    cur_timet = time(&cur_timet);
    strftime(time_str, 29, "%X", localtime(&cur_timet));

    get_prefix(line_prefix);

    fprintf(logtpcc ? logtpcc : ERROUT, "%s %s - ", line_prefix,time_str);
    vfprintf(logtpcc ? logtpcc : ERROUT, format, ap);
    if (logtpcc)
        fflush(logtpcc);

    if (debug && logtpcc) {
        fprintf(ERROUT, "%s %s - ", line_prefix,time_str);
        vfprintf(ERROUT, format, ap);
    }

    va_end(ap);
}

void prefix_sprintf(char*buf, char *format, ...)
{
    time_t cur_timet;
    char time_str[30];
    char line_prefix[50];
    char info[256];
    va_list ap;

    va_start(ap, format);

    cur_timet = time(&cur_timet);
    strftime(time_str, 29, "%X", localtime(&cur_timet));

    get_prefix(line_prefix);

    sprintf(buf, "%s %s - ", line_prefix,time_str);
    vsprintf(info, format, ap);
    strcat(buf,info);

    va_end(ap);
}
```

client_utils.h

```
/*
 *
 *                  client_utils.h
 *
 * $Revision: 1.11 $
 * $Date: 1999/05/06 21:28:26 $
 * $Log:          $
 *
 *
```

```
*
*
* $TALog: client_utils.h,v$
* Revision 1.11  1999/05/06  21:28:26  oz
* - Removed all the .. from the includes
* - Added -I.. to the makefiles instead
* - Moved all the thread related code and connection
*   selection to serverMon.c
* [from r1.8 by delta oz-24309-TPCC-add-oracle8.1-code, r1.5]
*
* Revision 1.8  1998/12/14  20:27:54  wenjian
* Made corresponding changes due to data structure change of tran_info_t.
*
* - Change data structure tran_info_t
* [from r1.7 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.3]
*
* Revision 1.7  1998/12/11  16:14:19  wenjian
* Add code for checking statistic data in a single variable and collecting
* statistic data based on iStatsFrequency.
*
* - Add total_num_trans to tran_info_t;
* [from r1.6 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.1]
*
* Revision 1.6  1998/11/09  16:59:37  wenjian
* In this revision, most of the changes are related to the directory of header
* files after directory reorganization. Other changes include adding or removing
* files to put them in the right directories. Makefiles are written for NT
* platform so that nmake is working on NT now. Need a top level Makefile for all
* the directories.
* [from r1.5 by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.2]
*
* Revision 1.5  1998/11/09  14:48:15  wenjian
* In an effort to make a new directory structure for TPCC, this delta
* creates two directories: tpcc/client and tpcc/server. All the files
* for this revision are copied from tpcc/sp-tpcc without any change.
* Further change may be needed for some files due to the change of
* the directory structure.
* [added by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.1]
*
* Revision 1.4  1998/11/09  14:26:51  wenjian
* Change enc_status to a data structure that has fields:
*   - Status code
*   - Line Number
*   - File Name
*   - Encina Error Code
*   - Error Msg
* Remove statusMsgs in web_tpcc.c
*
* Add definition of enc_status_t
* [from r1.19 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.6]
*
* Revision 1.19  1998/10/22  21:13:07  wenjian
* [merge of changes from 1.11 to 1.18 into 1.14]
*
* Revision 1.18  1998/10/22  19:18:32  dongfeng
* [from r1.17 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.2]
*
* Revision 1.17  1998/10/08  14:18:00  dongfeng
* Add codes for doing web-based tpcc.
* [from r1.11 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.1]
*
* Revision 1.14  1998/08/18  14:38:39  wenjian
* Minor change
* [from r1.13 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.4]
*
* Revision 1.13  1998/08/18  13:35:42  wenjian
* Remove NUM_NEXT_REPORTS since it is no use.
* [from r1.11 by delta wenjian-22495-TPCC-add-new-feature-to-monitor-tpcc-clients, r1.7]
*
* Revision 1.11  1998/06/17  15:28:51  wenjian
* Add 'double time' in struct total_tran_count_t.
* [from r1.10 by delta wenjian-22495-TPCC-add-new-feature-to-monitor-tpcc-clients, r1.2]
*
* Revision 1.10  1998/04/29  19:47:43  wenjian
* - Define ENDMSG marking the end of socket message between tpcc_client
*   and tpcc_monitor
* - Remove ENCINA_C_CALLING_CONVENTION from err_printf
* [from r1.9 by delta wenjian-22495-TPCC-add-new-feature-to-monitor-tpcc-clients, r1.1]
*
* Revision 1.9  1998/02/17  22:13:41  wenjian
* [merge of changes from 1.6 to 1.7 into 1.8]
*
* Revision 1.7  1998/02/17  16:04:41  oz
* - Split the login into two parts to allow for special logins
* - If the warehouse ID is 0, this is a special login to
*   query the client for status
* [from r1.6 by delta oz-21864-TPCC-split-client-login-screen, r1.1]
*
* Revision 1.8  1998/02/17  22:07:00  wenjian
* Minor changes for NT
* [from r1.6 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.1]
*
* Revision 1.6  1998/01/26  16:43:32  oz
* - Removed the code for collecting stats in the client
*   and dumping them before exit.
*
* - Removed timeP and time_allocated from thread_info_t
* [from r1.5 by delta oz-21691-TPCC-remove-client-stats-code, r1.1]
*
* Revision 1.5  1998/01/26  16:19:23  oz
* - moved all the code pertaining to the background
*   thread to its own file and all the data structures
*   to client_utils.h
* [from r1.4 by delta oz-21689-TPCC-move-client-bg-thread-to-separate-file, r1.1]
*
* Revision 1.4  1998/01/23  15:07:47  oz
* - Updated the SP TPCC directory to the latest files used
*   during the SP tpcc audit.
* [from r1.3 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
*
*
*    client_utils.h
*                Generic utilities used by the client processes
*/

#ifndef TPCC_CLIENT_UTILS_H
#define TPCC_CLIENT_UTILS_H

#include "common/tpcc_type.h"
#include <stdio.h>
#include <time.h>
#include <encina/encina.h>
#include "client.h"
#ifdef WIN32
#include <winsock.h>
#endif
#include <tpm/mon/mon.h>

/*
 *  err_printf
 *                Print a string to stderr after prefixing it with the client
 *                info and the current time.
 *  logprintf
 *                Prints as above to the log file.
 */
#ifdef WEB_TPCC_CLIENT
extern FILE * errtpcc;
#endif
extern FILE *logtpcc;
extern char log_file_name[];
extern void logprintf( char *format, ...);
extern void err_printf( char *format, ...);
extern void prefix_sprintf( char *buf, char *format, ...);

/*  tran_timing_t: for debug:
 *  Keep track of the timestamps of all the transactions
 *  and dump it out upon exit.  There is an array of timestamps
 *  per thread and each thread dumps it when it exits.
 */
typedef struct {
  int server;
  int terminal;
  int tran;
  int sub_tran;  /*  Subclass: for NewOrder and payment: 1=>hasRemote  */
  struct timeval start;                 /*  Time received from terminal */
  struct timeval send;                  /*  Time the RPC was made (explicit only)*/
  struct timeval srvr_start;      /*  Time received by server     */
  struct timeval srvr_done;       /*  Time sent by server         */
  struct timeval end;                   /*  Time sent to terminal       */
  int num_rms;                          /*  Number of RMs the tran involved */
  int tran_failed;
} tran_timing_t;

typedef enum {
   thread_state_init = 0,
   thread_state_called,
   thread_state_sent,
   thread_state_received,
   thread_state_returned,
   thread_state_done
} thread_state_t;

#define NUM_STATES thread_state_done
#define ENDMSG "...."  /* a special string to mark the end of a message */

typedef struct {
   thread_state_t state;
   int tran;
   struct timeval init, called, sent, received, returned, done;
   int printed, done_printed;
} thread_descr_t;

typedef struct {
    int num;
    int errs;
    double RTtotal[2];
    int RTcount;
} tran_info_t;
/*
 * total_tran_count_t
 *
 * structure that holds the total count of transaction of each type
 * as well as the reposne times.
 *
 */
typedef struct {
    tran_info_t tran[MAX_TRAN_TYPE + 1];
    int errors;
```

```
    double time;   /* used for tools/tpcc_monitor.c */
} total_tran_count_t;

/* enc_status_t
 * structure that holds error information
 */
typedef struct {
    int status;
    int line;
    char file[268];
    unsigned long encinaError;
    char errorMsg[ENCINA_MAX_STATUS_STRING_SIZE];
} enc_status_t;

/*
 *  thread_info_t
 *
 *  per thread information kept by this module
 */
typedef struct {
    int thread_index;
    int thread_id;
    int initialized;
    tran_timing_t last_tran;
    int num_trans;
    int consecutive_errors;
    thread_descr_t descr;
    tran_info_t tran[MAX_TRAN_TYPE + 1];
    int done;
} thread_info_t;

int  time_diff_ms(struct timeval *t2, struct timeval *t1);

extern int debug;
#define DPRINT(args)  if (debug) err_printf args

extern MUTEX_T init_lock;
extern int info_list_len;
extern thread_info_t **info_list;  /*  List of all the thread info */

/*
 *  A global variable by which the process would like to
 *  identify itself in the prefix to output
 */
extern int user_id;

/*
 *  clientUtils_ReportReturn
 *                  Called when a transaction is returned in order to error codes
 */
extern void clientUtils_ReportReturn(char *msg, data_header *statusP);

#define CHECK_ENVIRON(str,var)  if (str == NULL) { fprintf(ERROUT, \
                                    "%s environment variable is not defined.\n",var);  }

char *clientUtils_TranCodeToName(int type);

#endif /*  TPCC_CLIENT_UTILS_H  */
```

databuf.h

```
/*
 *     databuf.h
 *
 * $Revision: 1.1 $
 * $Date: 1998/11/06 21:10:11 $
 * $Log:          databuf.h,v $
 * Revision 4.2  95/05/16  10:55:31  10:55:31  tpcc (TPCC Benchmark)
 * Added necessary RCS ident strings
 *
 * Revision 4.1  95/05/09  15:21:02  15:21:02  strue (Scott Truesdale)
 * New code from Transarc - initial version
 *
 * Revision 3.2  95/04/03  17:43:09  17:43:09  strue (Scott Truesdale)
 * Changes from Transarc - added sql error handling in client; cleaned up debug handling with macros; added
check on db paramters via call to server.
 *
 * Revision 3.1  95/04/03  15:10:30  15:10:30  strue (Scott Truesdale)
 * Base of rev 3 - shipped to transarc
 *
 *
 *
 *
 *  $TALog: databuf.h,v $
 * Revision 1.1  1998/11/06  21:10:11  dongfeng
 * - Move all files common to client and server to tpcc/common
 *   directory
 * [added by delta dongfeng-23677-TPCC-new-directory-structures, r1.1]
 *
 * Revision 1.3  1998/10/22  15:33:04  wenjian
 * Make changes to Encina server code to connect with SQL server and add
 * callsql.c and sql directory.
 *
 * Add ERR_BAD_ITEM_ID, which is returned by SLQnew and same as INVALID_NEWO
 * [from r1.2 by delta wenjian-23529-TPCC-integrate-with-SQL-server, r1.1]
 *
 * Revision 1.2  1998/01/23  15:07:47  oz
 * - Updated the SP TPCC directory to the latest files used
 *   during the SP tpcc audit.
 * [from r1.1 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
 *
 * Revision 1.1  1997/04/20  11:57:57  oz
 * - This is the code base modified at IBM Poughkeepsie
 *   by Ofer Zajicek and Radha Sivaramakrishnan for the
 *   SP scaling test for TPCC.
 * [added by delta oz-19782-TPCC-add-ibm-sp-code, r1.1]
 *
 * Revision 1.31  1995/10/30  19:10:54  oz
 * [merge of changes from 1.29 to 1.30 into 1.27]
 *
 * Revision 1.30  1995/10/27  15:41:30  oz
 * - Modified the tpc-c code to work with the new informix
 *   sql code that is in ex_trans.ec
 * [from r1.29 by delta oz-16761-TPCC-modify-code-to-work-with-oracle, r1.1]
 *
 * Revision 1.27  1995/10/20  18:44:30  ctipper
 * [merge of changes from 1.17 to 1.25 into 1.22]
 *
 * Revision 1.25  1995/10/20  18:15:34  ctipper
 * Incorporate changes per code review.
 *
 * - add DISTRIBUTED_TRAN_FAILED, TPCC_DB_INFO_PARTIAL, and
 * TPCC_DB_INFO_FAILED error codes to  tpcc_rc_t
 * - got rid of MAX_NUM_SERVERS variables
 * [from r1.23 by delta ctipper-16547-TPCC-more-distributed-trans, r1.2]
 *
 * Revision 1.23  1995/10/13  17:00:26  ctipper
 * This delta encompasses all changes necessary to do distributed,XA
 * transactions with the TPCC benchmark. This includes the changes
 * necessary to build with Informix version 6.
 *
 * Each client still talks to only one server, however, if a distributed
 * transaction is necessary, the client sends the request to a different
 * interface of that server which then forwards all or part of the
 * request on to the appropriate remote server.
 *
 * - added new error codes to the  tpcc_rc_t enumeration.
 * - defined MAX_NUM_SERVERS to be 10
 * [from r1.19 by delta ctipper-16547-TPCC-more-distributed-trans, r1.1]
 *
 * Revision 1.19  1995/09/20  21:02:39  oz
 * -Corrected code for the payment transaction
 * - The distributed case now no longer uses
 *   stored procedures
 * [from r1.18 by delta oz-16547-TPCC-add-distributed-transactions, r1.2]
 *
 * Revision 1.18  1995/09/20  17:51:10  oz
 * - Added distributed transactions for the new order and
 *   payment transaction
 *
 * - Added new error codes
 * [from r1.17 by delta oz-16547-TPCC-add-distributed-transactions, r1.1]
 *
 * Revision 1.22  1995/10/02  20:31:07  oz
 * - Corrected definition of ERROR()
 * [from r1.21 by delta oz-16638-tpcc-modify-terminal-for-RTE, r1.3]
 *
 * Revision 1.21  1995/10/02  18:51:45  oz
 * - Added definitions needed for utils.c and liberty.c
 * [from r1.20 by delta oz-16638-tpcc-modify-terminal-for-RTE, r1.2]
 *
 * Revision 1.20  1995/10/02  15:52:35  oz
 * - Modified the TPC-C benchmark to be compatible with the RTE.
 * - There are now 3 terminal processes:
 *     emulator: the old terminal process with a built in
 *                     simple emulator
 *     curses: An interactive terminal process using curses
 *     liberty: An interactive terminal process to be used with
 *                     the RTE compatible with the liberty freedom terminal.
 *
 * - Define TRUE and FALSE only if they are not already defined.
 *   (curses.h defines TRUE)
 * - Removed READ_TO_DATE and YEAR_TO_SECOND
 * - Added term_type_t
 * - Added
 *                 GOOD_INPUT (0)
 *                 WRONG_INPUT (10)
 * [from r1.17 by delta oz-16638-tpcc-modify-terminal-for-RTE, r1.1]
 *
 * Revision 1.17  1995/07/28  15:28:23  oz
 * - Added a -null and -no_marshalling option to TPCC
 *
 * - Added INVALID_TRAN_TYPE return code
 * [from r1.16 by delta oz-16070-TPCC-add-null-and-marshalling-test, r1.1]
 *
 * Revision 1.16  1995/07/18  17:02:38  oz
 * - Added a DCE_ERROR error code
 * [from r1.15 by delta oz-15938-TPCC-add-dce-only-client, r1.1]
 *
 * Revision 1.15  1995/05/22  19:50:48  shl
 * [merge of changes from 1.12 to 1.13 into 1.14]
 *
 * Revision 1.13  1995/05/18  15:11:27  oz
 * [from r1.12 by delta oz-15290-TPCC-incorporate-hp-drop-of-05-16-95, r1.1]
 *
 * Revision 1.14  1995/05/22  17:26:35  ctipper
 * [merge of changes from 1.5 to 1.9 into 1.11]
```

```
*
* [*** log entries omitted ***]
*
*/

#ifndef __TPCC_DATABUF_H__
#define __TPCC_DATABUF_H__

#define I_NAME_LEN      24
#define I_DATA          50
#define W_NAME_LEN      10
#define ADDR_LEN        20
#define STATE_LEN       2
#define ZIP_LEN         9
#define DIST_INFO_LEN   24
#define S_DATA_LEN      50
#define D_NAME_LEN      10
#define H_DATA_LEN      24
#define CARRIER_LEN     2
#define C_LAST_LEN          17
#define C_MID_LEN       2
#define PHONE_LEN       16
#define CREDIT_LEN      2
#define C_DATA_LEN      500
#define BC_DTA_LEN          23

#define YEAR_TO_DATE   1
#define YEAR_TO_SECOND 2

#define ERROR(x) fprintf(stderr,"Error: %s\n",#x),exit(11)

#define MAX_STR_LEN     255
#define MAX_OL                                  15

#ifndef TRUE
#define TRUE    1
#endif
#ifndef FALSE
#define FALSE   0
#endif

#define CANCEL                          -1

#define DATETIME_LEN    19

#define D_PER_W         10

#define COLLECTOR           1                                   /* ctipper 5/3/95 */

#define ERR_BAD_ITEM_ID  1  /* copied from sql/tpcc.h */
#define RPC_ERROR                               -2
#define SUCCESS_CODE                            0

#define CHAR_NULL       '\0'                    /* strue 1/23/95 */

typedef enum {
   liberty_term,
   curses_term,
   emulator_term
} term_type_t;

typedef enum {
   TPCC_SUCCESS = 0,
   GOOD_INPUT = 0,

   INVALID_NEWO = 100,
   SQL_ERROR = 2,
   TRPC_ERROR = 3,
   DCE_ERROR = 4,
   NO_SUCH_LAST_NAME = 5,
   INVALID_TRAN_TYPE = 6,
   INVALID_HANDLE = 7,

   WRONG_INPUT = 10,

   DISTRIBUTED_TRAN_FAILED = 15,

   TPCC_DB_INFO_PARTIAL = 20,
   TPCC_DB_INFO_FAILED,

   TPCC_ERROR_BEGIN_NEWO = 110,

   TPCC_ERROR_DECL_NEWO_SEL_ITEM,
   TPCC_ERROR_OPEN_NEWO_SEL_ITEM,
   TPCC_ERROR_OPEN_DIST_NEWO_SEL_ITEM,
   TPCC_ERROR_FETCH_NEWO_SEL_ITEM,
   TPCC_ERROR_FETCH_DIST_NEWO_SEL_ITEM,
   TPCC_ERROR_PREP_NEWO_SEL_STCK,
   TPCC_ERROR_DECL_NEWO_SEL_STCK,
   TPCC_ERROR_OPEN_NEWO_SEL_STCK,
   TPCC_ERROR_OPEN_DIST_NEWO_SEL_STCK,
   TPCC_ERROR_FETCH_NEWO_SEL_STCK,
   TPCC_ERROR_FETCH_DIST_NEWO_SEL_STCK,
   TPCC_ERROR_NEWO_SELECT,
   TPCC_ERROR_NEWO_UPD_STCK,
   TPCC_ERROR_DIST_NEWO_UPD_STCK,
   TPCC_ERROR_NEWO_SELECT_2,
   TPCC_ERROR_DECL_NEWO_SEL_CUST,
   TPCC_ERROR_OPEN_NEWO_SEL_CUST,
   TPCC_ERROR_OPEN_DIST_NEWO_SEL_CUST,
   TPCC_ERROR_FETCH_NEWO_SEL_CUST,
   TPCC_ERROR_FETCH_DIST_NEWO_SEL_CUST,
   TPCC_ERROR_DECL_NEWO_SEL_DIST,
   TPCC_ERROR_OPEN_NEWO_SEL_DIST,
   TPCC_ERROR_OPEN_DIST_NEWO_SEL_DIST,
   TPCC_ERROR_FETCH_NEWO_SEL_DIST,
   TPCC_ERROR_FETCH_DIST_NEWO_SEL_DIST,
   TPCC_ERROR_PREP_NEWO_INS_OL,
   TPCC_ERROR_DECL_NEWO_INS_OL,
   TPCC_ERROR_OPEN_NEWO_INS_OL,
   TPCC_ERROR_OPEN_DIST_NEWO_INS_OL,
   TPCC_ERROR_PUT_NEWO_INS_OL,
   TPCC_ERROR_PUT_DIST_NEWO_INS_OL,
   TPCC_ERROR_DECL_NEWO_SEL_WARE,
   TPCC_ERROR_OPEN_NEWO_SEL_WARE,
   TPCC_ERROR_OPEN_DIST_NEWO_SEL_WARE,
   TPCC_ERROR_FETCH_NEWO_SEL_WARE,
   TPCC_ERROR_FETCH_DIST_NEWO_SEL_WARE,
   TPCC_ERROR_EXECUTE_NEWO_UPD_INS,
   TPCC_ERROR_UPDATE_NEWO_NEXT_OID,
   TPCC_ERROR_PREP_NEWO_INS,
   TPCC_ERROR_EXECUTE_DIST_NEWO_INS,
   TPCC_ERROR_EXECUTE_NEWO_COMMIT,
   TPCC_ERROR_ROLLBACK_NEWO,
   TPCC_ERROR_REMOTE_OL_SELECT,
   TPCC_ERROR_REMOTE_OL_UPDATE,

   TPCC_ERROR_OPEN_ORDS_CNT_CID = 200,
   TPCC_ERROR_FETCH_ORDS_CNT_CID,
   TPCC_ERROR_OPEN_ORDS_SEL_CLAST,
   TPCC_ERROR_FETCH_ORDS_SEL_CLAST,
   TPCC_ERROR_OPEN_ORDS_SEL_CID,
   TPCC_ERROR_FETCH_ORDS_SEL_CID,
   TPCC_ERROR_OPEN_ORDS_SEL_OLDORD,
   TPCC_ERROR_FETCH_ORDS_OLDORD,
   TPCC_ERROR_OPEN_ORDS_SEL_OL,
   TPCC_ERROR_FETCH_ORDS_SEL_OL,
   TPCC_ERROR_EXECUTE_ORDS_COMMIT,

   TPCC_ERROR_OPEN_DELIVERY_OLDEST_OID = 300,
   TPCC_ERROR_FETCH_DELIVERY_OLDEST_OID,
   TPCC_ERROR_EXECUTE_DELIVERY_COMMIT,
   TPCC_ERROR_OPEN_DELIVERY_SEL_ORD,
   TPCC_ERROR_FETCH_DELIVERY_SEL_ORD,
   TPCC_ERROR_OPEN_DELIVERY_SEL_SUM_OL,
   TPCC_ERROR_FETCH_DELIVERY_SEL_SUM_OL,
   TPCC_ERROR_EXECUTE_DELIVERY_EXEC_DVRY,
   TPCC_ERROR_SELECT_DELIVERY_ORDER_ID,
   TPCC_ERROR_SELECT_DELIVERY_CARRIER_ID,
   TPCC_ERROR_SELECT_DELIVERY_BALANCE,

   TPCC_ERROR_OPEN_STOCKLEVEL_SEL_OID = 400,
   TPCC_ERROR_FETCH_STOCKLEVEL_SEL_OID,
   TPCC_ERROR_OPEN_STOCKLEVEL_CNT_SID,
   TPCC_ERROR_FETCH_STOCKLEVEL_CNT_SID,
   TPCC_ERROR_OPEN_STOCKLEVEL_FIND,
   TPCC_ERROR_FETCH_STOCKLEVEL_FIND,
   TPCC_ERROR_EXECUTE_STOCKLEVEL_COMMIT,

   TPCC_ERROR_OPEN_PAYMENT_CNT_CID = 500,
   TPCC_ERROR_FETCH_PAYMENT_CNT_CID,
   TPCC_ERROR_OPEN_PAYMENT_SEL_CLAST,
   TPCC_ERROR_FETCH_PAYMENT_SEL_CLAST,
   TPCC_ERROR_OPEN_PAYMENT_SEL_CID,
   TPCC_ERROR_FETCH_PAYMENT_SEL_CID,
   TPCC_ERROR_DECL_PAYMENT_SEL_DIST,
   TPCC_ERROR_OPEN_PAYMENT_SEL_DIST,
   TPCC_ERROR_OPEN_DIST_PAYMENT_SEL_DIST,
   TPCC_ERROR_FETCH_PAYMENT_SEL_DIST,
   TPCC_ERROR_FETCH_DIST_PAYMENT_SEL_DIST,
   TPCC_ERROR_DECL_PAYMENT_SEL_WARE,
   TPCC_ERROR_OPEN_PAYMENT_SEL_WARE,
   TPCC_ERROR_OPEN_DIST_PAYMENT_SEL_WARE,
   TPCC_ERROR_FETCH_PAYMENT_SEL_WARE,
   TPCC_ERROR_FETCH_DIST_PAYMENT_SEL_WARE,
   TPCC_ERROR_EXECUTE_PAYMENT_UPD_CUST_LAST,
   TPCC_ERROR_EXECUTE_PAYMENT_UPD_CUST_ID,
   TPCC_ERROR_COMMIT_PAYMENT_UPD_CUST,
   TPCC_ERROR_SELECT_PAYMENT_W_YTD,
   TPCC_ERROR_SELECT_PAYMENT_D_YTD,
   TPCC_ERROR_BEGIN_PAYMENT,
   TPCC_ERROR_EXECUTE_PAYMENT_COMMIT,
   TPCC_ERROR_PAYMENT_UPD_CUST_BY_NAME,
   TPCC_ERROR_PAYMENT_UPD_CUST_BY_ID,
   TPCC_ERROR_PAYMENT_UPDATE_DIST,
   TPCC_ERROR_PAYMENT_UPDATE_WH,
   TPCC_ERROR_PAYMENT_INSERT_HISTORY,
   TPCC_ERROR_EXECUTE_PAYMENT_WH_DIST

} tpcc_rc_t;

typedef enum {
   TPCC_DEADLOCK_MSG = 10,
   TPCC_RETRY_MSG
} tpcc_msg_t;

#endif /* __TPCC_DATABUF_H__ */
```

debug.c

```
#include <stdio.h>
#include <string.h>
#include <stdarg.h>
#include <sys/stat.h>
#include <errno.h>
#include <math.h>
#include <time.h>
#include <fcntl.h>
#include <unistd.h>
#ifdef WIN32
#include <process.h>
#else
#include <termio.h>
#endif

int print_thread_id = 0;
int user_id = 1;
char *user_code = "C";

int get_thread_id()
{
   return(0);
}

/*
 *  get_prefix
 *                  Format the output prefix for printing:
 *                  It contains the user_id, 'C' or 'T' depending on whether it
 *                  is a terminal or a client and optional a thread identifier
 *                  The prefix is written in the buffer passed in by the caller.
 */
void get_prefix(buffer)
 char *buffer;
{
   if (print_thread_id) {
                  int thread_id = get_thread_id();
                  sprintf(buffer,  "%s(%d-%s-%d)%s",
                                    user_id < 10 ? "   " : user_id < 100 ? " " : "",
                                    user_id,
                                    user_code,
                                    thread_id,
                                    thread_id < 10 ? " " : "");
   } else {
                  sprintf(buffer,  "%s(%2d-%s)",
                                    user_id < 10 ? " " : "", user_id, user_code);
   }
}


/*
 * err_printf
 *                  A var-arg function that appends the current time and
 *                  other data to the print request and sends it to  stderr
 */
void err_printf(char *format, ...)
{
   static int initialized = 0;
   static FILE *debug_f = NULL;
   time_t cur_timet;
   char time_str[30];
   char line_prefix[50];
   va_list ap;

   va_start(ap, format);

   if (!initialized) {
                  char fileName[45];
                  initialized = 1;
                  sprintf(fileName,"DebugFile.%d", getpid());
                  debug_f = fopen(fileName,"w");
   }

   cur_timet = time(&cur_timet);
   strftime(time_str, 29, "%X", localtime(&cur_timet));

   get_prefix(line_prefix);

   if (debug_f) {
                  fprintf(debug_f, "%s %s - ", line_prefix,time_str);
                  vfprintf(debug_f, format, ap);
                  fflush(debug_f);
   }

   va_end(ap);
}

void set_client_debug_state(void *contextP, int state, int tran)
{
}
```

delivery.tacf

```
/*
 * Copyright (C) 1991, 1990 Transarc Corporation
 * All Rights Reserved
 */
/*
 *  neworder.tacf -- attribute configuration file for tpcc server.
 * used for transparent binding
 *
 * $Revision: 1.1 $
 * $Date: 1998/11/06 21:10:11 $
 * $Log:          tpcc.tacf,v $
 *
 * $TALog: delivery.tacf,v$
 * Revision 1.1  1998/11/06  21:10:11  dongfeng
 * - Move all files common to client and server totpcc/common
 *   directory
 * [added by delta dongfeng-23677-TPCC-new-directory-structures, r1.1]
 *
 * Revision 1.1  1997/04/20  11:57:57  oz
 * - This is the code base modified at IBM Poughkeepsie
 *   by Ofer Zajicek and Radha Sivaramakrishnan for the
 *   SP scaling test for TPCC.
 * [added by delta oz-19782-TPCC-add-ibm-sp-code, r1.1]
 *
 * Revision 1.3  1996/01/12  16:06:44  oz
 * - Added transaction specific servers:  there are 5 different interfacese
 *   one for each transaction type.
 * [added by delta oz-16955-TPCC-add-transaction-specific-servers, r1.1]
 *
 */

[implicit_handle(mon_handle_t handle)]
interface delivery
{
}
```

delivery.tidl

```
/*
 * id: $id: $
 *
 * component_name: encina benchmarks
 *
 * the following functions list may not be complete.
 * functions defined by/via macros may not be included.
 *
 * functions:
 *    <fill_me_in>
 *
 * origins: transarc corp.
 *
 * (c) copyright transarc corp. 1995, 1993
 * all rights reserved
 * licensed materials - property of transarc
 *
 * us government users restricted rights - use, duplication or
 * disclosure restricted by gsa adp schedule contract with transarc corp
 */
/*
 * history
 * $talog: $
 */

/*
 * delivery.tidl-- interface definition file fortpccserver.
 *
 * $revision: 1.11 $
 * $date: 1995/10/20 21:55:05 $
 * $log:          tpcc.tidl,v $
 */

[uuid(d714d8f8-2105-11cf-830f-0800093b9834), version(1.0)]

interface delivery
{
import "tpm/mon/mon_handle.idl";
import "tpcc_type.idl";

[nontransactional] void
                  impTPCCDelivery([in,out]delivery_data_t *dataP,
                                                [out] trpc_status_t  * trpcStatus);

}
```

do_tpcc.c

```
/*
 *      do_tpcc.c
 *
 * $Revision: 1.12 $
 * $Date: 1999/05/06 21:28:26 $
 * $Log:          do_tpcc.c,v $
 *
 * $TALog: do_tpcc.c,v $
 * Revision 1.12  1999/05/06  21:28:26  oz
 * - Removed all the .. from the includes
 * - Added -I.. to the  makefiles instead
 * - Moved all the thread related code and connection
 *   selection to serverMon.c
 * [from r1.8 by delta oz-24309-TPCC-add-oracle8.1-code, r1.5]
```

```
*
* Revision 1.8  1999/01/29  20:16:32  wenjian
* Call init_encina_clientsince client_inithas been renamed to
* init_encina_client
* [from r1.7 by delta wenjian-23787-TPCC-integrate-code-for-AIX-and-NT, r1.7]
*
* Revision 1.7  1998/12/09  14:44:53  wenjian
* Add a call to client_init() in main
* [from r1.6 by delta wenjian-23787-TPCC-integrate-code-for-AIX-and-NT, r1.3]
*
* Revision 1.6  1998/11/09  16:59:37  wenjian
* In this revision, most of the changes are related to the directory of header
* files after directory reorganization. Other changes include adding or removing
* files to put them in the right directories. Makefiles are written for NT
* platform so that nmake is working on NT now. Need a top level Makefile for all
* the directories.
* [from r1.5 by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.2]
*
* Revision 1.5  1998/11/09  14:48:15  wenjian
* In an effort to make a new directory structure for TPCC, this delta
* creates two directories: tpcc/client and tpcc/server. All the files
* for this revision are copied from tpcc/sp-tpcc without any change.
* Further change may be needed for some files due to the change of
* the directory structure.
* [added by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.1]
*
* Revision 1.15  1998/02/17  22:07:01  wenjian
* Define macros to deal with the different function names on NT
* [from r1.14 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.1]
*
* Revision 1.14  1998/01/26  20:37:34  oz
* - Remove all the code associated with explicit binding
*
* - Removed bindingType
* - Removed client_first_wh and client_last_wh
* - Removed command line args: binding, offset, ware
* [from r1.13 by delta oz-21697-TPCC-remove-explicit-binding-code, r1.1]
*
* Revision 1.13  1998/01/26  15:33:31  oz
* - Changed default binding to transparent
* [from r1.12 by delta oz-21671-TPCC-merge-online-transaction-interfaces, r1.2]
*
* Revision 1.12  1998/01/24  14:17:05  oz
* - User server name to identify server and name delivery file
* - Use env variable HOME instead of /home/encina if HOME is set
* [from r1.11 by delta oz-21687-TPCC-use-server-name-to-identify-process, r1.1]
*
* Revision 1.11  1998/01/23  15:07:48  oz
* - Updated the SP TPCC directory to the latest files used
*   during the SP tpcc audit.
* [from r1.10 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
*
* Revision 1.8  1997/08/04  19:50:41  oz
* [from r1.7 by delta oz-20506-TPCC-convert-to-new-format-of-connection-manager, r1.2]
*
*
*
*/
/*====================================================================
* do_tpcc.c
*
*
*  This is the main client program for the TPCC benchmark using Encina.
*
*  The client program is multi-threaded: there is one thread for each
*  terminal and one thread to process incoming connections.
*
*  When the client starts up it starts a listening thread.  That thread
*  calls the encina function cnm_ManageConnections and provides it a
*  port number.  The client spawns a thread (through cnm_ManageConnections)
*  for each terminal that connects to it.  That thread receives the input
*  from the terminal and translates it to a transaction data structure
*  (such as payment_data_t or newOrder_data_t).  The terminal thread
*  (in the client) then sends an RPC over to the server to process the request.
*

          ____________________________
          |       Client             |
          |       Process            |              ----------
------   RPC  |                       |   socket      | Terminal |
     |<------|-------------------Terminal-----~~~~~~-------|          |
     |     |                 Thread   |              ----------
Server|     |                       |
     |     |                       |              ----------
     |  RPC  |                       |   socket      | Terminal |
------ <------|-------------------Terminal-----~~~~~~-------|          |
          |                 Thread   |              ----------
          |                       |
          |___________________________|


*
*
*====================================================================
*/



#include <stdio.h>
#include <string.h>
#include <stdarg.h>
#include <sys/stat.h>
#include <errno.h>
#include <math.h>
#include <time.h>
#include <fcntl.h>
#ifndef WIN32
#include <termio.h>
#endif
#include <unistd.h>
#include <sys/ipc.h>
#include <tpm/mon/mon_client.h>
#include <tc/tc.h>
#include <dce/rpc.h>

#if defined (solaris)
#include <dce/pthread.h>
#else   /* solaris */
#include <pthread.h>
#endif

#include "common/tpcc_type.h"
#include "common/utilities.h"
#include "client_utils.h"
#include "common/do_tpcc.h"
#include "client_listen.h"
#include "client.h"

extern char *sys_errlist[];  /*  Translations of errno file errors */

#ifdef WIN32
#define STRCMP _stricmp
#else
#define STRCMP strcasecmp
#endif

/*
 *  ENTERING
 *              A macro that is called before processing a TPCC transaction.
 *              If debug mode is enabled it prints a message containing the name
 *              of the transaction being executed.
 */
#define ENTERING(msg) if (debug) err_printf("Entering %s\n", msg)

#define MAX_CONSECUTIVE_ERRS 3000
/*
 *  RETURNED
 *              A macro that is called after a transaction has been processed.
 *              If the transaction failed it reports an error.
 *              In debug mode it also emits a message indicating the processing
 *              has been completed.
 */
#define RETURNED(msg, hdrP)                             \
{                                                       \
  if (((hdrP)->returncode == TPCC_SUCCESS) ||           \
      ((hdrP)->returncode == INVALID_NEWO)) {           \
    consecutiveErrors = 0;                              \
  } else {                                              \
    consecutiveErrors++;                                \
  }                                                     \
  if (debug ||                                          \
      (((hdrP)->returncode != SUCCESS_CODE) &&          \
       ((hdrP)->returncode != INVALID_NEWO))) {         \
    clientUtils_ReportReturn(msg, hdrP);                \
    if (consecutiveErrors > MAX_CONSECUTIVE_ERRS) {     \
      err_printf("Too many consecutive errors (%d)\n",   \
                                        consecutiveErrors);             \
      exit_program(1);                                  \
    }                                                   \
  }                                                     \
}


int                             useSecurity = FALSE;
int                             null_test = 0;
int                             client_lock_handles= 0;

/* The following are global to the client*/
char *LOG_FILE_DIR = "runs/threads";
int             user_id;
int             user_port = 4011;
char    *user_code = "C";       /*  Prefix for output to identify this
                                                                        *  process as a client or a terminal
                                                                        */
int             consecutiveErrors = 0;
char   *result_dir;

int debug = 0;
char log_file_name[100];
int logtrans = 0;
FILE *logtpcc = NULL;

static void check_parms(int argc, char *argv[]);
static void print_header(int argc, char *argv[]);


/*=====================================================*/

main(argc, argv)
```

```
int argc;
char *argv[];
{
   check_parms(argc,argv);  /*  Read and parse the command line parameters */

   err_printf("Client %d starting.\n", user_id);

   init_encina_client(user_id);
   enroll_client(user_id); /* enroll as a client */

   /*
    *  Open log file
    */
   logtpcc = fopen(log_file_name,"w");
   print_header(argc, argv);   /** Print a test header to the logfile **/

   /*
    *    Start the listening thread:
    *    This call will not return
    */
   make_connections((void *)user_port);

   exit_program( 0 );
   return (0);                      /* to statisfy lint */

}      /***************       end of main     *************/

/*=========================================================*/

/*
 * User must supply user_id as a parm and all other parameters
 * as environment variables.
 */

/*-----------------------------------------------------------------
 *       Check Parameters
 *  Check the parameters passed in.
 *
 *  Not all the parameters are relevant for this executable.
 *  This code is shared between the regular Encina Monitor
 *  based TPC-C client and other test clients that do not
 *  use the Encina Monitor.  The type of this executable is
 *  in client_type and is set to mon_client for the TPCC
 *  Monitor based client (the audited client).
 *----------------------------------------------------------------
 */
static void check_parms (argc,argv)
  int argc;
  char *argv[];

{
   char *host_name = getenv("HOST");
   char *home_dir = getenv("HOME");
   int              next_arg = 1;
   int errors = 0;
   char             *progName;
   int print_help = 0;

   user_id = -1;
   result_dir = ".";

   while (next_arg < argc) {
                  if (!STRCMP("-debug", argv[next_arg])) {
                     /*  Enable debug mode (for testing)  */
                     debug = 1;
                  } else if (!STRCMP("-dir", argv[next_arg])) {
                     /*  The directory for the client output  */
                     result_dir = argv[++next_arg];
                  } else if (!STRCMP("-log", argv[next_arg])) {
                     /*  A less intrusive form of debug mode  */
                     logtrans = 1;
                  } else if (!STRCMP("-id", argv[next_arg])) {
                     /*  The id of this client  */
                     user_id = atol(argv[++next_arg]);
                  } else if (!STRCMP("-port", argv[next_arg])) {
                     /*  The id of this client  */
                     user_port = atol(argv[++next_arg]);
                     if (user_id < 0) user_id = user_port;
                  } else if (!STRCMP("-security", argv[next_arg])) {
                     /*  Enable security between the client and the server.
                      *  This is enabled by default
                      */
                     useSecurity = TRUE;
                  } else if (!STRCMP("-noSecurity", argv[next_arg])) {
                     /*  Disable security between the client and the server.
                      *  This is enabled by default
                      */
                     useSecurity = FALSE;
                  } else if (!STRCMP("-null", argv[next_arg])) {
                     /*  For testing: do not access the data in the DB  */
                     logprintf("Performing NULL test\n");
                     null_test = 1;
                  } else if (!STRCMP("-lock", argv[next_arg])) {
                     logprintf("Locking longterm handles\n");
                     client_lock_handles= atol(argv[++next_arg]);
                  } else {
                     printf("invalid parameter: %s\n", argv[next_arg]);
                     print_help = 1;
                     break;
                  }
               next_arg++;
   }

   if (user_id < 0) {
     printf("  Missing User Id\n");
     print_help = 1;
   }

   if (print_help) {
                  progName = strrchr(argv[0], '/');
                  progName = (progName ? progName + 1 : argv[0]);

                  printf("\nusage:\n You can specify the following in anyorder\n");
                  printf(" You must specify the Id\n");

                  printf("   -id <num>      The user ID for this client\n");
                  printf("   -dir <dir>     Directory for output (default \".\")\n");
                  printf("   -debug         enable debugging\n");
                  printf("   -log           log all activity to a file\n");
                  printf("   -security      enable secure communications between the client and PA\n");
                  printf("   -null          NULL test: the server immediately returns\n");

                  exit(-1);
   }

   sprintf(log_file_name,"%s/%s/C.%s.%d",
                     home_dir ? home_dir : "/home/encina",
                     LOG_FILE_DIR,
                     host_name ? host_name : "host", user_id);
}

/*
 *  print_header:
 *                Print some feedback to the user on the client configuration
 */
static void print_header(int argc, char *argv[])
{
   int i;
   if (!logtpcc)
                  return;

   logprintf("Client %d starting a %s test.\n",
                     user_id,
                     null_test ? "NULL" : "DB");

   logprintf("Params: ");

   for (i=0; i<argc; i++) {
                  fprintf(logtpcc, "%s ", argv[i]);
   }
   fprintf(logtpcc, "\n");
   fflush(logtpcc);
}
```

do_tpcc.h

```
/*
 * do_tpcc.h
 *
 *  $Revision: 1.1 $
 *  $Date: 1998/11/09 16:00:05 $
 *  $Log:         do_tpcc.h,v $
 *
 *  $TALog: do_tpcc.h,v $
 * Revision 1.1  1998/11/09  16:00:05  dongfeng
 * Move do_tpcc.h to common directory
 * [added by delta dongfeng-23677-TPCC-new-directory-structures, r1.4]
 *
 * Revision 1.7  1998/01/23  15:07:49  oz
 * - Updated the SP TPCC directory to the latest files used
 *   during the SP tpcc audit.
 * [from r1.6 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
 *
 *
 */

#ifndef _DO_TPCC_H_INCLUDED_
#define _DO_TPCC_H_INCLUDED_

#include <dce/rpc.h>
#include <trpc/trpc.h>
#include "databuf.h"

#define WRONG_INPUT      0
#define NEW_ORDER        1
#define PAYMENT                          2
#define ORDER_STATUS     3
#define DELIVERY         4
#define STOCK_LEVEL      5
#define QUIT             9
#define MIN_OL                           5
#define MAX_FLDS       200      /* Maximum fields in a TPC-C form */

#define THRESHOLD_LEN   2

#define ON               1
#define OFF              0
```

```
#define YES    1
#define NO     0

#define INSIZE  1024

#define DO_ROLLBACK     1
#define DONT_ROLLBACK   0

/**  The response time requirements for the transactions in seconds.
 **  90% of the transactions are required to have a response time less
 **  than or equal to the value below.
 **/
#define NEWORD_90RT      5
#define PAYMENT_90RT     5
#define ORDSTAT_90RT     5
#define DELIVERY_90RT    5  /* 5 for interactive or 80 for background */
#define STOCKLEV_90RT   20

/*
 *  What type of client is this?
 */
typedef enum {
   tk_client,
   dce_client,
   mon_client,
   db_client
} client_type_t;

extern client_type_t client_type;

typedef enum {
   transparent, explicit, longTerm, noReservation
} binding_t;

/* Handle from client to PA is now described using both thepaHandle
and the mondHandle. */

#define NUM_TRANS        5
#define NEWO_ERR         6
#define PAYMENT_ERR      7
#define ORD_STAT_ERR     8
#define DELIVERY_ERR     9
#define STOCK_ERR       10
#define NEWO_ROLLBACK   11
#define END_OF_WINDOW  0xff
#define BEGIN_WINDOW   0xaa
#ifndef SHORT_WAITS
#define NEWO_MEAN_THINK_TIME                 122
#define PAYMENT_MEAN_THINK_TIME              122
#define ORDER_STAT_MEAN_THINK_TIME           102
#define DELIVERY_MEAN_THINK_TIME              51
#define STOCK_MEAN_THINK_TIME                 51
#define NEWO_MIN_KEY_TIME                    185
#define PAYMENT_MIN_KEY_TIME                  31
#define ORDER_STAT_MIN_KEY_TIME               21
#define DELIVERY_MIN_KEY_TIME                 21
#define STOCK_MIN_KEY_TIME                    21
#else
#define NEWO_MEAN_THINK_TIME                  61
#define PAYMENT_MEAN_THINK_TIME               61
#define ORDER_STAT_MEAN_THINK_TIME            51
#define DELIVERY_MEAN_THINK_TIME              26
#define STOCK_MEAN_THINK_TIME                 26
#define NEWO_MIN_KEY_TIME                     93
#define PAYMENT_MIN_KEY_TIME                  16
#define ORDER_STAT_MIN_KEY_TIME               11
#define DELIVERY_MIN_KEY_TIME                 11
#define STOCK_MIN_KEY_TIME                    11
#endif


#endif /* _DO_TPCC_H_INCLUDED_ */
```

encina.C

```
/* (C)1997 IBM Corporation */
/*****************************************************************/
/*
*/
/*                  File: tuxclient.h                            */
/*
*/
/*****************************************************************/

#include <stdlib.h>
#include "inout.h"
#include "encina.h"

extern "C" {
}

extern "C" send_new_order(void *contextP, NewOrder_data *dataP);
extern "C" send_payment(void *contextP, Payment_data *dataP);
extern "C" send_stock_level(void *contextP, StockLevel_data *dataP);
extern "C" send_order_status(void *contextP, OrderStatus_data *dataP);
extern "C" send_delivery(void*contextP, Delivery_data *dataP);

void Encina::cleanup() {
}

Encina::Encina() {
   return;
}

Encina::~Encina() {
   return;
}

int Encina::tran (NewOrder_data *dataP, void *contextP, char *servname) {
   send_new_order(contextP, dataP);
   return 0;
}


int Encina::tran (Payment_data *dataP, void *contextP, char *servname) {
   send_payment(contextP, dataP);
   return 0;
}


int Encina::tran (OrderStatus_data *dataP, void *contextP, char *servname) {
   send_order_status(contextP, dataP);
   return 0;
}


int Encina::tran (StockLevel_data *dataP, void *contextP, char *servname) {
   send_stock_level(contextP, dataP);
   return 0;
}

int Encina::tran (Delivery_data *dataP, void *contextP, char *servname) {
   send_delivery(contextP, dataP);
   return 0;
}

int Encina::tran (char *servname) {
   return -1;
}

int Encina::atran (char *servname) {

   return 0;
}
```

encina.h

```
/* (C)1997 IBM Corporation */
/*****************************************************************/
/*
*/
/*                  File: tuxclient.h                            */
/*
*/
/*****************************************************************/

#ifndef ENCINA_H
#define ENCINA_H

const int TMINBUFSIZE = 1536;

class Encina {
public:
   static void cleanup();
   int tran(char *servname);
   int tran(NewOrder_data *dataP, void *contextP, char *servname);
   int tran(Payment_data *dataP, void *contextP, char *servname);
   int tran(StockLevel_data *dataP, void *contextP, char *servname);
   int tran(OrderStatus_data *dataP, void *contextP, char *servname);
   int tran(Delivery_data *dataP, void *contextP, char *servname);
   int atran(char *servname);
   Encina();
   ~Encina();
};

extern Encina encina;

#endif
```

encina_client.c

```
/*
 *                  encina_client.c
 *
 * $Revision: 1.7 $
 * $Date: 1999/05/06 21:28:26 $
 * $Log:            $
 *
 *
 * $TALog: encina_client.c,v$
 * Revision 1.7  1999/05/06  21:28:26  oz
 * - Removed all the .. from the includes
 * - Added -I.. to the  makefiles instead
```

```
* - Moved all the thread related code and connection
*   selection to serverMon.c
* [from r1.6 by delta oz-24309-TPCC-add-oracle8.1-code, r1.5]
*
* Revision 1.6  1998/11/09  16:59:37  wenjian
* In this revision, most of the changes are related to the directory of header
* files after directory reorganization. Other changes include adding or removing
* files to put them in the right directories. Makefiles are written for NT
* platform so that nmake is working on NT now. Need a top level Makefile for all
* the directories.
* [from r1.5 by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.2]
*
* Revision 1.5  1998/11/09  14:48:16  wenjian
* In an effort to make a new directory structure for TPCC, this delta
* creates two directories: tpcc/client and tpcc/server. All the files
* for this revision are copied from tpcc/sp-tpcc without any change.
* Further change may be needed for some files due to the change of
* the directory structure.
* [added by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.1]
*
* Revision 1.5  1998/01/23  15:07:51  oz
* - Updated the SP TPCC directory to the latest files used
*   during the SP tpcc audit.
* [from r1.4 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
*
*
*
*/

/*
*  encina_client.c
*
*               The Encina related code in the client that is common to both
*               the monitor client and the toolkit client.
*
*/

#include <stdio.h>
#include <string.h>
#include <stdarg.h>
#include <trpc/trpc.h>
#include <encina/encina.h>
#include "common/utilities.h"
#include "client_utils.h"
#include "encina_client.h"


static trpc_handle_t bind_to_server(char *name);

/*
*  encina_error_message
*
*  Report an encina error message by interpreting it and writing
*  it to both the logfile (if any) and to standard error
*/
void encina_error_message(msg, n)
 char *msg;
 unsigned long n;
{
   char errorMsg[ENCINA_MAX_STATUS_STRING_SIZE];
   encina_StatusToString(n, ENCINA_MAX_STATUS_STRING_SIZE, errorMsg);
   err_printf("ERROR: %s.  Error code = %s (%d  0x%x) \n", msg, errorMsg, n, n);
}

/*
*  encina_error
*
*  This is called for FATAL errors.  It reports the error and exits.
*/
void encina_error(funcName, n)
 char *funcName;
 unsigned long n;
{
   char msg[128];
   sprintf("%s failed", funcName);
   encina_error_message(msg, n);
   exit_program(1);
}

/*
*   secure_handle
*
*               Secure a handle to an encina server.
*               This can be called with either a PA handle or with
*               a trpc handle to a toolkit server.
*/
void secure_handle(trpc_handle_t handle, int use_security)
{
   rpc_binding_handle_t       rpcHandle;
   unsigned long              status = 0;
   unsigned char              *serverPrincipal;

   ENCINA_CALL("trpc_GetRpcHandleFromBinding",
            trpc_GetRpcHandleFromBinding(handle,&rpcHandle));

   rpc_mgmt_inq_server_princ_name(rpcHandle,rpc_c_authn_default,
                           &serverPrincipal, &status);

   if (use_security) {
               DPRINT(("rpc_binding_set_auth_info-> principal %s, protect %d, authn %d authz %d\n",
                           serverPrincipal, rpc_c_protect_level_connect,
                           rpc_c_authn_default, rpc_c_authz_dce));

               rpc_binding_set_auth_info(rpcHandle,serverPrincipal,
                                                   rpc_c_protect_level_connect,
                                                   rpc_c_authn_default,
                                                   NULL,
                                                   rpc_c_authz_dce,
                                                   &status);
   } else {
               DPRINT(("rpc_binding_set_auth_info-> principal %s, protect %d, authn %d authz %d\n",
                           serverPrincipal, rpc_c_protect_level_none,
                           rpc_c_authn_default, rpc_c_authz_dce));

               rpc_binding_set_auth_info(rpcHandle,serverPrincipal,
                                                   rpc_c_protect_level_none,
                                                   rpc_c_authn_default,
                                                   NULL,
                                                   rpc_c_authz_dce,
                                                   &status);
   }


   if (status != rpc_s_ok) {
      switch (status) {
         case rpc_s_invalid_binding:
            printf("rpc binding invalid****** \n");
            break;
         case rpc_s_wrong_kind_of_binding:
            printf("rpc binding is the wrong kind \n");
            break;
         case rpc_s_unknown_authn_service:
            printf("rpc authn service unknown \n");
            break;
      } /* switch */
      bde_Exit(1);
   }
}
```

encina_client.h

```
/*
*               encina_client.h
*
* $Revision: 1.5 $
* $Date: 1998/11/09 14:48:16 $
* $Log:             $
*
*
* $TALog: encina_client.h,v$
* Revision 1.5  1998/11/09  14:48:16  wenjian
* In an effort to make a new directory structure for TPCC, this delta
* creates two directories: tpcc/client and tpcc/server. All the files
* for this revision are copied from tpcc/sp-tpcc without any change.
* Further change may be needed for some files due to the change of
* the directory structure.
* [added by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.1]
*
* Revision 1.5  1998/01/23  15:07:52  oz
* - Updated the SP TPCC directory to the latest files used
*   during the SP tpcc audit.
* [from r1.4 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
*
*
*    Declarations common to monitor version and toolkit version
*
*/

#ifndef ENCINA_CLIENT_H
#define ENCINA_CLIENT_H

#include <trpc/trpc.h>

#define CELL_NAME_UNAVAILABLE                   1
#define MON_RETRIEVEENABLE_FAILED               2
#define MON_INITCLIENT_FAILED                   3
#define MON_SECURITYSET_FAILED                  4
#define MON_SETREFRESHINTERVAL_FAILED           5
#define NOINFO_TRPC_ERROR                       6

void encina_error_message(char *msg, unsigned long n);
void encina_error(char *funcName, unsigned long n);
void secure_handle(trpc_handle_t handle, int use_security);


#endif  /*  ENCINA_CLIENT_H  */
```

field.C

```
/* (C)1997 IBM Corporation */
#include <stdio.h>
#include "field.h"
#include "inout.h"
#include "format.h"
```

```
#if 0
#if USE_ALLOCA
#include <alloca.h>
#endif
#endif

extern char const * const blanks;
extern char const * const underscores;
extern char const * const backspaces;
extern int position(InOut *ioP, int x, int y);

Field *genfield(InOut *ioP, int x, int y, int len, int *ptr) {
   return new IntField(ioP, x, y, len, ptr);
}
Field *genfield(InOut *ioP, int x, int y, int len, short *ptr) {
   return new ShortField(ioP, x, y, len, ptr);
}
Field *genfield(InOut *ioP, int x, int y, int len, long *ptr) {
   return new LongField(ioP, x, y, len, ptr);
}
Field *genfield(InOut *ioP, int x, int y, int len, char *ptr) {
   return new TextField(ioP, x, y, len, ptr);
}
Field *genfield(InOut *ioP, int x, int y, int len, double *ptr) {
   return new MoneyField(ioP, x, y, len, ptr);
}
Field *genfield(InOut *ioP, int x, int y, int len, unsigned char *ptr) {
   return new Int8Field(ioP, x, y, len, ptr);
}

/*****************************************************************************
Field
*****************************************************************************/
Field::Field(InOut *inoutP, int size, char *str)
   : ioP(inoutP), len(size), pos(0), changed(0), need_redisplay(0)
{
   need_free_string = need_free = 0;
   if (str == NULL) {
                string = new char[len+1];
                need_free_string = 1;
   } else {
                string = str;
   }
   ok_func = NULL;
   ok_data = NULL;
   string[0] = 0;
}

Field::Field(InOut *ioP, int inx, int iny, int size, char *str)
   : ioP(ioP), x(inx), y(iny), len(size), pos(0), changed(0), need_redisplay(0)
{
   need_free_string = need_free = 0;
   if (str == NULL) {
                string = new char[len+1];
                need_free_string = 1;
   } else {
                string = str;
   }
   ok_func = NULL;
   ok_data = NULL;
   string[0] = 0;
}

int Field::reset() {
   pos=0;
   changed=0;
   return 0;
}
Field::~Field() {
   if (need_free_string)
                delete [] string;
}
int Field::finalize_field(){
   changed = 0;
   string[pos] = 0;
   return 0;
}
int Field::display_field(int use_underscores) {
   position(ioP, x,y);
   ioP->write(string);
   if (use_underscores) {
                ioP->write(underscores, len-pos);
   } else {
                ioP->write(blanks, len-pos);
   }
   return 0;
}
int Field::get_key() {
   char key;
   int cc;
   cc = ioP->read(&key, 1);

   return (cc == 0)  ? EOF : key ;
}
int Field::add_char(int key) {
   if (pos >= len || (!isprint(key) && key != ' ')) {
                ioP->write("\a", 1);
                return 1;
   }
   changed = 1;
   string[pos] = key;
   ioP->write(&string[pos++], 1);
   return 0;
}
int Field::backspace() {
   ioP->write("\b_\b", 3);
   changed = 1;
   pos--;
   return 0;
}
int Field::start_position() {
   position(ioP, x, y);
   return 0;
}
int Field::get_field(int need_pos) {
   int key;

   if (need_pos)
                position(ioP, x, y);
   if (pos != 0) {
                need_redisplay = 1;
                ioP->write(string, pos);
                ioP->write(underscores, len-pos);
                if (len-pos < 6)
                   ioP->write(backspaces, len-pos);
                else
                   position(ioP, x+pos, y);
   }

   ioP->mark();
   while (1) {
                key = get_key();
                switch(key) {
                case EOF:
                   return EOF;

                case '\r':   /* Carriage Return */
                case '\n':   /* Newline */
                   ioP->hold();
                   if (changed) {
                                finalize_field();
                   }
                   ioP->pop();
                   display_field(1);
                   return ENTER;
                   break;

                case '\t':   /* Tab */
                case '\006': /* Ctrl-F */
                case '\016': /* Ctrl-N */
                   if (changed) {
                                finalize_field();
                   }
                   ioP->pop();
                   display_field(1);
                   return NEXT_FIELD;
                   break;

                case '\002': /* Ctrl-B */
                case '\020': /* Ctrl-P */
                   if (changed) {
                                finalize_field();
                   }
                   ioP->pop();
                   display_field(1);
                   return PREV_FIELD;

                case '\b':   /* Backspace */
                case '\177': /* Del */
                   if (pos > 0) {
                                backspace();
                   } else
                                ioP->write("\a", 1);
                   break;

                case '\014': /* Ctrl-L */
                   ioP->pop();
                   return REDISPLAY;

                case '\030': /* Ctrl-X */
                case '\003': /* Ctrl-C */
                   ioP->unmark();
                   return ABORT;

                default:
                   add_char(key);
                }
   }
}

/*****************************************************************************
IntField
*****************************************************************************/
IntField::IntField(InOut *ioP, int inx, int iny, int size, int *val) : Field(ioP, inx, iny, size), value(val) {
   if (value==NULL) {
                value = new int;
                need_free=1;
   }
```

```
}
IntField::IntField(InOut *ioP, int size, int *val) : Field(ioP, size), value(val) {
   if (value==NULL) {
                value = new int;
                need_free=1;
   }
}
IntField::~IntField() {
   if (need_free)
                delete value;
}

int IntField::add_char(int key) {
   if (pos < len && isdigit(key)) {
                changed = 1;
                string[pos] = key;
                ioP->write(&string[pos++], 1);
                return 0;
   }
   ioP->write("\a", 1);
   return 1;
}
int IntField::display_field(int use_underscores) {
   int firstchar;
#if USE_ALLOCA
   char *buf = (char *)alloca(len+1);
#else
   char *buf = new char[len+1];
#endif
   memset(buf, 'x', len);
   if (pos)
                firstchar = format_int(buf, len+1, *value);
   else
                firstchar = len;
   position(ioP, x, y);
   if (use_underscores) {
                ioP->write(underscores, firstchar);
                ioP->write(buf+firstchar, len-firstchar);
   } else {
                ioP->write(buf, len);
   }
   return 0;
}
int IntField::finalize_field(){
   changed = 0;
   string[pos] = 0;
   if (value != NULL)
                *value = atoi(string);
   return 0;
}

/************************************************************************
ShortField
************************************************************************/
ShortField::ShortField(InOut *ioP, int inx, int iny, int size, short *val) : Field(ioP, inx, iny, size), value(val) {
   if (value==NULL) {
                value = new short;
                need_free=1;
   }
}
ShortField::ShortField(InOut *ioP, int size, short *val) : Field(ioP, size), value(val) {
   if (value==NULL) {
                value = new short;
                need_free=1;
   }
}
ShortField::~ShortField() {
   if (need_free)
                delete value;
}

int ShortField::add_char(int key) {
   if (pos < len && isdigit(key)) {
                changed = 1;
                string[pos] = key;
                ioP->write(&string[pos++], 1);
                return 0;
   }
   ioP->write("\a", 1);
   return 1;
}
int ShortField::display_field(int use_underscores) {
   int firstchar;
#if USE_ALLOCA
   char *buf = (char *)alloca(len+1);
#else
   char *buf = new char[len+1];
#endif
   if (pos)
                firstchar = format_short(buf, len+1, *value);
   else
                firstchar = len;
   position(ioP, x, y);
   if (use_underscores) {
                ioP->write(underscores, firstchar);
                ioP->write(buf+firstchar, len-firstchar);
   } else {
                ioP->write(buf, len);
   }
   return 0;
}
int ShortField::finalize_field(){
   changed = 0;
   string[pos] = 0;
   if (value != NULL)
                *value = atoi(string);
   return 0;
}

/************************************************************************
ShortField
************************************************************************/
Int8Field::Int8Field(InOut *ioP, int inx, int iny, int size, unsigned char *val) : Field(ioP, inx, iny, size),
value(val) {
   if (value==NULL) {
                value = new unsigned char;
                need_free=1;
   }
}
Int8Field::Int8Field(InOut *ioP, int size, unsigned char *val) : Field(ioP, size), value(val) {
   if (value==NULL) {
                value = new unsigned char;
                need_free=1;
   }
}
Int8Field::~Int8Field() {
   if (need_free)
                delete value;
}

int Int8Field::add_char(int key) {
   if (pos < len && isdigit(key)) {
                changed = 1;
                string[pos] = key;
                ioP->write(&string[pos++], 1);
                return 0;
   }
   ioP->write("\a", 1);
   return 1;
}
int Int8Field::display_field(int use_underscores) {
   int firstchar;
#if USE_ALLOCA
   char *buf = (char *)alloca(len+1);
#else
   char *buf = new char[len+1];
#endif
   if (pos)
                firstchar = format_char(buf, len+1, *value);
   else
                firstchar = len;
   position(ioP, x, y);
   if (use_underscores) {
                ioP->write(underscores, firstchar);
                ioP->write(buf+firstchar, len-firstchar);
   } else {
                ioP->write(buf, len);
   }
   return 0;
}
int Int8Field::finalize_field(){
   changed = 0;
   string[pos] = 0;
   if (value != NULL)
                *value = atoi(string);
   return 0;
}

/************************************************************************
LongField
************************************************************************/
LongField::LongField(InOut*ioP, int inx, int iny, int size, long *val) : Field(ioP, inx, iny, size), value(val) {
   if (value==NULL) {
                value = new long;
                need_free=1;
   }
}
LongField::LongField(InOut*ioP, int size, long *val) : Field(ioP, size), value(val) {
   if (value==NULL) {
                value = new long;
                need_free=1;
   }
}
LongField::~LongField(){
   if (need_free)
                delete value;
}
int LongField::add_char(int key) {
   if (pos < len && isdigit(key)) {
                changed = 1;
                string[pos] = key;
                ioP->write(&string[pos++], 1);
                return 0;
   }
   ioP->write("\a", 1);
   return 1;
}
int LongField::display_field(int use_underscores) {
```

```
    int firstchar;
#if USE_ALLOCA
    char *buf = (char *)alloca(len+1);
#else
    char *buf = new char[len+1];
#endif
    if (pos)
                firstchar = format_long(buf, len+1, *value);
    else
                firstchar = len;
    position(ioP, x, y);
    if (use_underscores) {
                ioP->write(underscores, firstchar);
                ioP->write(buf+firstchar, len-firstchar);
    } else {
                ioP->write(buf, len);
    }
    return 0;
}
int LongField::finalize_field(){
    changed = 0;
    string[pos] = 0;
    if (value != NULL)
                *value = atoi(string);
    return 0;
}

/*****************************************************************************
MoneyField
*****************************************************************************/
MoneyField::MoneyField(InOut*ioP, int inx, int iny, int size, double *val) : Field(ioP, inx, iny, size),
value(val) {
    seen_dollar = seen_sign = seen_dot = seen_digit = 0;
    if (value==NULL) {
                value = new double;
                need_free=1;
    }
}
MoneyField::MoneyField(InOut*ioP, int size, double *val) : Field(ioP, size), value(val) {
    seen_dollar = seen_sign = seen_dot = seen_digit = 0;
    if (value==NULL) {
                value = new double;
                need_free=1;
    }
}
MoneyField::~MoneyField(){
    if (need_free)
                delete value;
}
int MoneyField::add_char(intkey) {
    do {
                if (pos >= len)
                    break;
                if (key == '$') {
                    if (!(pos == 0 || (pos == 1 && seen_sign))) break;
                    seen_dollar = 1;
                } else if (key == '-') {
                    if (!(pos == 0 || (pos == 1 && seen_dollar))) break;
                    seen_sign = 1;
                } else if (key == '.') {
                    if (seen_dot) break;
                    seen_dot = 1;
                } else if (!isdigit(key))
                    break;
                if (seen_dot) {
                    if (seen_dot >= 4)
                                break;
                    seen_dot++;
                }
                changed = 1;
                string[pos] = key;
                ioP->write(&string[pos++], 1);
                return 0;
    } while (0);
    ioP->write("\a", 1);
    return 1;
}
int MoneyField::backspace() {
    ioP->write("\b_\b", 3);
    changed = 1;
    pos--;
    if (seen_dot)
                seen_dot--;
    if (string[pos] == '-')
                seen_sign = 0;
    if (string[pos] == '$')
                seen_dollar = 0;
    if (string[pos] == '.')
                seen_dot = 0;
    return 0;
}
int MoneyField::display_field(intuse_underscores) {
    int firstchar;
#if USE_ALLOCA
    char *buf = (char *)alloca(len+1);
#else
    char *buf = new char[len+1];
#endif
    if (pos)
                firstchar = format_money(buf,len+1, *value);
    else
                firstchar = len;
    position(ioP, x, y);
    if (use_underscores) {
                ioP->write(underscores, firstchar);
                ioP->write(buf+firstchar, len-firstchar);
    } else {
                ioP->write(buf, len);
    }
    return 0;
}
int MoneyField::finalize_field(){
    changed = 0;
    string[pos] = 0;
    if (value != NULL) {
                *value = atof(string + seen_dollar + seen_sign);
                if (seen_sign)
                    *value = -*value;
    }
    return 0;
}
int MoneyField::reset() {
    Field::reset();
    seen_dollar = seen_sign = seen_dot = seen_digit = 0;
    return 0;
}

/*****************************************************************************
TextField
*****************************************************************************/
TextField::TextField(InOut*ioP, int inx, int iny, int size, char *str) : Field(ioP, inx, iny, size, str) {
    value='TextField::string;
}
TextField::TextField(InOut*ioP, int size, char *str) : Field(ioP, size, str) {
    value='TextField::string;
}
int TextField::add_char(intkey) {
    if (pos >= len || (!isalnum(key)&& key != ' ' && key != '.')) {
                ioP->write("\a", 1);
                return 1;
    }
    changed = 1;
    string[pos] = key;
    ioP->write(&string[pos++], 1);
    return 0;
}
```

field.h

```
/* (C)1997 IBM Corporation */
#if !defined(INCLUDE_FIELD_H)
#define INCLUDE_FIELD_H

#include "inout.h"

class Field {
public:
    enum return_codes { INVALID, ENTER, NEXT_FIELD, PREV_FIELD, ABORT, REDISPLAY };
    InOut *ioP;
    int x, y;
    const int len;
    int pos;
    int changed;
    int need_redisplay;
    char *string;
    int (*ok_func)(void *data);
    int need_free;
    int need_free_string;
    void *ok_data;
    Field(InOut *ioP, int size, char *string=NULL);
    Field(InOut *ioP, int x, int y, int size, char *string=NULL);
    virtual ~Field();
    virtual int get_field (int need_pos=1);
    int get_key ();
    virtual int backspace();
    virtual int reset();
    virtual int start_position();
    virtual int add_char(int key);
    virtual int display_field(intuse_underscores=0);
    virtual int finalize_field();

    class Error {
                enum { USER_ABORT };
    };
};

class Int8Field : public Field {
public:
    unsigned char *value;
    int add_char(int key);
    int display_field(intuse_underscores=0);
    int finalize_field();

    Int8Field(InOut *ioP, int x, int y, int size, unsigned char *value=NULL);
    Int8Field(InOut *ioP, int size, unsigned char *value=NULL);
```

```
    virtual ~Int8Field();
};

class ShortField : public Field {
public:
    short *value;
    int add_char(int key);
    int display_field(intuse_underscores=0);
    int finalize_field();

    ShortField(InOut *ioP, int x, int y, int size, short *value=NULL);
    ShortField(InOut *ioP, int size, short *value=NULL);
    virtual ~ShortField();
};


class IntField : public Field {
public:
    int *value;
    int add_char(int key);
    int display_field(intuse_underscores=0);
    int finalize_field();

    IntField(InOut *ioP, int x, int y, int size, int *value=NULL);
    IntField(InOut *ioP, int size, int *value=NULL);
    virtual ~IntField();
};

class LongField : public Field {
public:
    long *value;
    int add_char(int key);
    int display_field(intuse_underscores=0);
    int finalize_field();

    LongField(InOut *ioP, int x, int y, int size, long *value=NULL);
    LongField(InOut *ioP, int size, long *value=NULL);
    virtual ~LongField();
};

class MoneyField: public Field {
public:
    int seen_dollar, seen_sign, seen_dot, seen_digit;
    double *value;
    int add_char(int key);
    int reset();
    int backspace();
    int display_field(intuse_underscores=0);
    int finalize_field();
    MoneyField(InOut *ioP, int x, int y, int size, double *value=NULL);
    MoneyField(InOut *ioP, int size, double *value=NULL);
    virtual ~MoneyField();
};

class TextField : public Field {
public:
    char *value;
    int add_char(int key);
    TextField(InOut *ioP, int x, int y, int size, char *value=NULL);
    TextField(InOut *ioP, int size, char *value=NULL);
};

Field *genfield(InOut *ioP, int x, int y, int len, int *ptr);
Field *genfield(InOut *ioP, int x, int y, int len, short *ptr);
Field *genfield(InOut *ioP, int x, int y, int len, long *ptr);
Field *genfield(InOut *ioP, int x, int y, int len, char *ptr);
Field *genfield(InOut *ioP, int x, int y, int len, unsigned char *ptr);
Field *genfield(InOut *ioP, int x, int y, int len, double *ptr);

#endif /* INCLUDE_FIELD_H */
```

format.C

```
/* (C)1997 IBM Corporation */
#include <string.h>
#include <math.h>

int format_char(char *buf, int size, char val) {
    int neg, pos;
    pos = size;
    buf[--pos] = 0;
    if (val == 0 && pos > 0) {
                buf[--pos] = '0';
                neg = 0;
    } else {
                neg = (val < 0) ? 1 : 0;
                if (neg) val = -val;
                while (val && pos > 0) {
                    buf[--pos] = (val % 10) + '0';
                    val /= 10;
                }
    }
    /* Too long */
    if (!pos && (val || neg)) {
                memset (buf, '*', size);
                return -1;
    }
    if (neg)
                buf[--pos] = '-';
    if (pos)
                memset(buf, ' ', pos);
    return pos;
}
int format_short(char *buf, int size, short val) {
    int neg, pos;
    pos = size;
    buf[--pos] = 0;
    if (val == 0 && pos > 0) {
                buf[--pos] = '0';
                neg = 0;
    } else {
                neg = (val < 0) ? 1 : 0;
                if (neg) val = -val;
                while (val && pos > 0) {
                    buf[--pos] = (val % 10) + '0';
                    val /= 10;
                }
    }
    /* Too long */
    if (!pos && (val || neg)) {
                memset (buf, '*', size);
                return -1;
    }
    if (neg)
                buf[--pos] = '-';
    if (pos)
                memset(buf, ' ', pos);
    return pos;
}
int format_int(char *buf, int size, int val) {
    int neg, pos;
    pos = size;
    buf[--pos] = 0;
    if (val == 0 && pos > 0) {
                buf[--pos] = '0';
                neg = 0;
    } else {
                neg = (val < 0) ? 1 : 0;
                if (neg) val = -val;
                while (val && pos > 0) {
                    buf[--pos] = (val % 10) + '0';
                    val /= 10;
                }
    }
    /* Too long */
    if (!pos && (val || neg)) {
                memset (buf, '*', size);
                return -1;
    }
    if (neg)
                buf[--pos] = '-';
    if (pos)
                memset(buf, ' ', pos);
    return pos;
}
int format_long(char *buf, int size, long val) {
    int neg, pos;
    pos = size;
    buf[--pos] = 0;
    if (val == 0 && pos > 0) {
                buf[--pos] = '0';
                neg = 0;
    } else {
                neg = (val < 0) ? 1 : 0;
                if (neg) val = -val;
                while (val && pos > 0) {
                    buf[--pos] = (val % 10) + '0';
                    val /= 10;
                }
    }
    /* Too long */
    if (!pos && (val || neg)) {
                memset (buf, '*', size);
                return -1;
    }
    if (neg)
                buf[--pos] = '-';
    if (pos)
                memset(buf, ' ', pos);
    return pos;
}

int format_float(char *buf, int size, int dec, double val) {
    static double pow10[] = { 1, 10, 100, 1000, 10000, 100000, 1000000 };
    int neg, pos;
    pos = size;
    buf[--pos] = 0;
#ifndef WIN32
    val = rint(val * pow10[dec]);
#else /* there is no rint on NT. Use floor instead */
    val = floor(val * pow10[dec]+0.5);
#endif
    neg = (val < 0) ? 1 : 0;
    if (neg) val = -val;

    while (val >= 1 && pos > 0) {
```

```
                if (!dec--) {
                    buf[--pos] = '.';
                    continue;
                }
                buf[--pos] = (int)fmod(val, 10) + '0';
                val /= 10;
    }
    if (dec >= 0) {
                while (dec >= 0 && pos > 0) {
                    if (!dec--) {
                                buf[--pos] = '.';
                    } else {
                                buf[--pos] = '0';
                    }
                }
                if (pos > 0)
                    buf[--pos] = '0';
    }
    /* Too long */
    if (!pos && (val >= 1 || neg)) {
                memset (buf, '*', size);
                return -1;
    }
    if (neg)
                buf[--pos] = '-';
    if (pos)
                memset(buf, ' ', pos);
    return pos;
}

int format_money(char *buf, int size, double val) {
    int pos;
    pos = format_float(buf, size, 2, val);
    if (pos > 0)
                buf[--pos] = '$';
    return pos;
}

int format_date(char *buf, int size, unsigned char* val) {
    memcpy(buf, val, size);
    buf[size]=0;
    return 0;
}

int format_phone(char *buf, int size, unsigned char *phone) {
    buf[0] = phone[0];
    buf[1] = phone[1];
    buf[2] = phone[2];
    buf[3] = phone[3];
    buf[4] = phone[4];
    buf[5] = phone[5];
    buf[6] = '-';
    buf[7] = phone[6];
    buf[8] = phone[7];
    buf[9] = phone[8];
    buf[10] = '-';
    buf[11] = phone[9];
    buf[12] = phone[10];
    buf[13] = phone[11];
    buf[14] = '-';
    buf[15] = phone[12];
    buf[16] = phone[13];
    buf[17] = phone[14];
    buf[18] = phone[15];
    buf[19] = '\0';
    return size;
}

int format_zip(char *buf, int size, unsigned char *zip) {
    buf[0] = zip[0];
    buf[1] = zip[1];
    buf[2] = zip[2];
    buf[3] = zip[3];
    buf[4] = zip[4];
    buf[5] = '-';
    buf[6] = zip[5];
    buf[7] = zip[6];
    buf[8] = zip[7];
    buf[9] = zip[8];
    buf[10] = '\0';
    return size;
}
```

format.h

```
/* (C)1997 IBM Corporation */
#if !defined(INCLUDE_FORMAT_H)
#define INCLUDE_FORMAT_H

int format_char (char *buf, int size, char   val);
int format_int  (char *buf, int size, int    val);
int format_long (char *buf, int size, long   val);
int format_short(char *buf, int size, short  val);
int format_float(char *buf, int size, int    dec, double val);
int format_money(char *buf, int size, double val);
int format_date (char *buf, int size, unsigned char  *val);
int format_phone(char *buf, int size, unsigned char  *phone);
int format_zip  (char *buf, int size, unsigned char  *zip);

#endif /* INCLUDE_FORMAT_H */
```

get_local_time.c

```
/*
 *                  get_local_time.c
 *
 * $Revision: 1.2 $
 * $Date: 1998/11/06 21:42:02 $
 * $Log:              $
 *
 *
 * $TALog: get_local_time.c,v $
 * Revision 1.2  1998/11/06  21:42:02  dongfeng
 * - Add makefile-nt
 *
 * - cast cur_t from double to long to get rid of some warnings.
 * [from r1.1 by delta dongfeng-23677-TPCC-new-directory-structures, r1.2]
 *
 * Revision 1.1  1998/11/06  21:10:12  dongfeng
 * - Move all files common to client and server totpcc/common
 *   directory
 * [added by delta dongfeng-23677-TPCC-new-directory-structures, r1.1]
 *
 * Revision 1.2  1998/10/22  19:18:33  dongfeng
 * [added by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.2]
 *
 */

#ifdef WIN32
#include "get_local_time.h"
#else
#include <sys/time.h>
#endif
#include <stdio.h>

#define Li2Double(x) ((double)((x).HighPart) * 4.294967296E9 + (double)((x).LowPart))

#ifdef WIN32
LARGE_INTEGER pFreq;
double sFreq;

get_time_init()
{
    QueryPerformanceFrequency(&pFreq);
    sFreq=Li2Double(pFreq);
}

get_local_time(struct timeval *timeP)
{
   double cur_t;
   LARGE_INTEGER counter;

   QueryPerformanceCounter(&counter);
   cur_t = Li2Double(counter) / sFreq;
   timeP->tv_sec = (long)cur_t;
   timeP->tv_usec = ((long)cur_t - timeP->tv_sec) * 1000000;
}

int gettimeofday(struct timeval *curTimeP, struct timezone *timezoneP)
{
   get_local_time(curTimeP);
   return 1;
}

#else
get_time_init()
{
}


get_local_time(struct timeval *timeP)
{
  struct timezone tz;

  gettimeofday(timeP, &tz);
}

#endif
```

get_local_time.h

```
#ifndef _GET_LOCAL_TIME_H_
#define _GET_LOCAL_TIME_H_

#ifdef WIN32
#include <windows.h>
#include <winsock.h>

/*
 * gettimeofday is not available in the MicrosoftC/C++ Run Time
 * and the Win32 API.
 */

*
 * It is not used and just for unix compatibility.
```

```
*/
struct timezone {
        char a;
};

get_time_init();

int gettimeofday(struct timeval *curTimeP, struct timezone *timezoneP);

#endif

#endif
```

inout.C

```
/* (C)1997 IBM Corporation */
#include <string.h>
#ifndef WIN32
#include <strings.h>
#endif
#include <unistd.h>
#include <stdlib.h>
#ifdef WIN32
#include <io.h>
#include <winsock.h>
#endif
#include <stdio.h>
#include <ctype.h>
#include <errno.h>

#include "screen.h"

extern char *sys_errlist[];

#if 1
void InOut::write(const void *buf, size_t size) {
   if (IOError) return;
   debug("write('%*.*s', %d);\n", size, size, buf, size);
   output.queue(buf, size);
   if (!Hold && input.len() == 0) { /* Don't write anything until there is no input*/
        flush();
   }
}

ssize_t InOut::read(void *buf, size_t size) {
   int rc;
   if (IOError) return(0);
   while (input.len() < size) {
#ifdef WIN32
        rc = recv(in_fd, (char *)input.ptr(), input.free(), 0);
#else
        rc = ::read(in_fd, input.ptr(), input.free());
#endif
      debug("::read('%*.*s', %d) = %d;\n", rc, rc, input.ptr(), input.free(), rc);
      if (inlog) {
          fwrite(input.ptr(), rc, 1, inlog);
          fflush(inlog);
      }
        if (rc > 0) {
          input.queue(rc);
        } else if (rc <= 0) {
          IOError = 1;
          return(0);
        }
   }
   memcpy(buf, input.ptr(), size);
   input.dequeue(size);
   debug("read('%*.*s', %d) = %d;\n", size, size, buf, size, size);
   return size;
}
#else
void InOut::write(const void *buf, size_t size) {
   debug("write('%s', %d);\n", buf, size);
#ifdef WIN32
   send(out_fd, (char *)buf, size,0);
#else
   ::write(out_fd, buf, size);
#endif
}

ssize_t InOut::read(void *buf, size_t size) {
   int rc;
   rc = ::read(in_fd, buf, size);
   debug("read('%s', %d) = %d;\n", buf, size, rc);
   return rc;
}
#endif

void InOut::flush() {
   debug("flush();\n");
   Hold = 0;
   if (IOError) return;
   while (output.len()) {
      debug("::write('%*.*s', %d);\n", output.len(), output.len(), output.ptr(), output.len());
#ifdef WIN32
        int rc = send(out_fd, (char *)output.ptr(), output.len(),0);
#else
        int rc = ::write(out_fd, output.ptr(), output.len());
#endif
      if (outlog) {
          fwrite(output.ptr(), rc, 1, outlog);
          fflush(outlog);
      }
        if (rc > 0) {
          output.dequeue(rc);
        } else if (rc < 0) {
          err_printf("Error writing data!\n");
          IOError = 1;
          return;
        }
   }
}

void InOut::write(const void *buf) {
   write(buf, strlen((const char *)buf));
}

InOut::InOut(int in, int out) : input(256), output(2048) {
#ifndef WIN32
   struct termios buf;
#endif

#ifdef DEBUG
   {
        char buf[256];
        sprintf(buf, "logs/debug.%d", getpid());
        debugfile = fopen(buf, "w");
        sprintf(buf, "logs/in.%d", getpid());
        inlog   = fopen(buf, "w");
        sprintf(buf, "logs/out.%d", getpid());
        outlog   = fopen(buf, "w");
   }
#endif

   int rc;
   Hold = 0;
   debugfile = inlog = outlog = (FILE *)0;
   IOError = 0;

   in_fd = in;
   if (out < 0)
        out_fd = in;
   else
      out_fd = out;
#ifndef WIN32
   if ((rc = tcgetattr(in_fd, &save_term)) < 0) {
        return;
   }

   buf = save_term;

   buf.c_lflag &= ~(ECHO | ICANON); /* echo off, canonical mode off */

   buf.c_cc[VMIN] = 1;     /* Case B: 1 byte at a time, no timer */
   buf.c_cc[VTIME] = 0;

   err_printf("echo off - tcsetattr on %d\n", in_fd);
   if (tcsetattr(in_fd, TCSAFLUSH, &buf) < 0)
        return;
#endif
}

InOut::~InOut() {
#ifdef WIN32
   return;
#else
   if (tcsetattr(in_fd, TCSAFLUSH, &save_term) < 0)
        return;
#endif
}
```

inout.h

```
/* (C)1997 IBM Corporation */

#ifndef INOUT_H
#define INOUT_H
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <ctype.h>
#ifndef WIN32
#include <termios.h>
#endif
#include <stdarg.h>
#include <string.h>

#include "tpcc.h"

/* This is for a VT100 */
#if 1
#define ESC "\033"
#define ESCc '\033'
#else
#define ESCc '^'
```

```
#define ESC "^"
#endif

#define TRIGGER "\021"
#define TRIGGERc '\021'

extern "C" err_printf(...);

#define POS(x,y) ESC "[" #y ";" #x "H"
#define CLEAR_EOS ESC "[J"

#ifdef WIN32
typedef int ssize_t;
#endif

class InOut {
private:
   class Buffer {
   private:
                int BufSize;
                enum { NUMMARKS=8 };
                char *buffer;
                int marks[NUMMARKS];
   public:
                int Pos;
                int Start;
      int num_marks;
                Buffer(int size) {
                   BufSize = size;
                   buffer = new char [BufSize];
                   Pos = Start = 0;
                   num_marks = 0;
                }
                int  pos()       { return Pos; };
                void pos(int P)   { Pos = P; };
                int  start()     { return Start; };
                void start(int S) { Start = S; };
                int  len()       { return Pos-Start; };
                int  free()      { return BufSize-Pos-1; };
                void *ptr()      { return &buffer[Start]; };
                int  lastmark()  { if (num_marks) return marks[num_marks-1]; return 999; };

                void  mark()     {
                   if (num_marks < NUMMARKS)
                                marks[num_marks++]= Pos;
                   else {
                                fprintf(stderr, "Buffer mark overflow\n");
                                exit (1);
                   }
                }
                void unmark() {
                   if (num_marks <= 0)
                                return;
                   num_marks--;
                }
                void pop() {
                   if (num_marks <= 0)
                                return;
                   if (marks[num_marks-1]>= Start) {
                                Pos=marks[--num_marks];
                   } else {
                                num_marks=0;
                   }
                }
                void queue(int size) {
                   Pos += size;
                }
                void queue(const void *buf, int size) {
                   /* If this is too big see if we can move what we have over*/
                   if (size+Pos >= BufSize) {
                                if (size + len() >= BufSize) {
                                   fprintf(stderr, "Buffer overflow\n");
                                   exit (1);
                                }
                                /* This requires memcpy to be "safe" */
                                if (Start + len() >= BufSize) {
                                   fprintf(stderr, "Strange Error: Start %d +  len %d >= size %d\n",
                                                Start ,  len(), BufSize);
                                   exit(1);
                                }
                                memcpy(buffer,&buffer[Start], len());
                                Pos -= Start;

                                /* Fix up our marks*/
                                int count = 0;
                                for (int i = 0; i < num_marks; i++) {
                                   if (marks[i] - Start >= 0)
                                                marks[count++] = marks[i] - Start;
                                }
                                num_marks = count;
                                Start = 0;
                   }
                   memcpy(&buffer[Pos],buf, size);
                   Pos += size;
                }
                void dequeue(int size) {
                   Start += size;
                   if (Start >= Pos) {
                                /* Fix up our marks*/
                                int count = 0;
                                for (int i = 0; i < num_marks; i++) {
                                   if (marks[i] - Start >= 0)
                                                marks[count++] = marks[i] - Start;
                                }
                                num_marks = count;

                                Start = Pos = 0;
                   }
                }
   };
   int in_fd, out_fd;
   int Hold;
#ifndef WIN32
   struct termios save_term;
#endif
   Buffer input;
   Buffer output;
   FILE *debugfile;
   FILE *inlog, *outlog;
public:
   int IOError;
   ssize_t read(void *buf, size_t size);
   void write(const void *buf, size_t size);
   void write(const void *buf);
   void flush();
   void mark() { debug("mark()\n"); output.mark(); };
   void unmark() { debug("unmark()\n"); output.unmark(); };
   void pop()  { debug("pop()\n\n"); output.pop(); };
   void hold() { debug("hold()\n"); Hold = 1; };
#ifdef DEBUG
   void debug(char *fmt, ...) {
                va_list args;

      fprintf(debugfile,"Start=%2d, Pos=%2d, Marks=%2d(%03d): ", output.Start,  output.Pos,
output.num_marks, output.lastmark());
                va_start(args,fmt);
                vfprintf (debugfile, fmt, args);
                va_end (args);
                ::fflush(debugfile);
   }
#else
   void debug(char *fmt, ...) {};
#endif
   InOut(int in=0, int out=1);
   ~InOut();
};

extern char const * const blanks;
extern char const * const underscores;
extern char const * const backspaces;

int format_int(char *buf, int size, int val);
int format_float(char *buf, int size, int dec, double val);
int format_money(char *buf, int size, double val);

#endif /*  INOUT_H */
```

mon_client.c

```
 /*
*                mon_client.c
*
* $Revision: 1.27 $
* $Date: 1999/05/26 16:29:52 $
* $Log:             $
*
*
* $TALog: mon_client.c,v $
*
* Change by Klavs Pedersen 2003/07/05 Explicit Binding to PA based on
*                        mapping file  .tpccbindrc
*
* Revision 1.27  1999/05/26  16:29:52  wenjian
* Sync with Austin code, and sync code for Oracle DB and SQL server.
* [from r1.26 by delta wenjian-24433-TPCC-clean-up-and-update, r1.2]
*
* Revision 1.26  1999/05/06  21:28:26  oz
* - Removed all the .. from the includes
* - Added -I.. to the  makefiles instead
* - Moved all the thread related code and connection
*   selection to serverMon.c
* [from r1.16 by delta oz-24309-TPCC-add-oracle8.1-code, r1.5]
*
* Revision 1.16  1999/01/29  20:16:33  wenjian
* - Rename client_init to init_encina_client because we have another
*   client_init in screen/client.C
* - Add code to read  StatsFrequency from .tpccrc (UNIX only)
* [from r1.15 by delta wenjian-23787-TPCC-integrate-code-for-AIX-and-NT, r1.7]
*
* Revision 1.15  1999/01/12  20:52:55  wenjian
* Call initialization function to create the shared file mapping between
* it and the corresponding dll.
* [from r1.14 by delta wenjian-23856-TPCC-integrate-with-NT-performance-monitor, r1.1]
```

```
*
* Revision 1.14  1998/12/28  20:07:12  wenjian
* - Change client_info to a pointer pClientInfo for flexibility.
* [from r1.13 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.5]
*
* Revision 1.13  1998/12/16  17:17:41  wenjian
* - Change (iStatsFrequency <= 1) to (iStatsFrequency < 1) in pre_rpc.
* [from r1.12 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.4]
*
* Revision 1.12  1998/12/14  20:27:54  wenjian
* Made corresponding changes due to data structure change of tran_info_t.
* [from r1.11 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.3]
*
* Revision 1.11  1998/12/11  16:14:19  wenjian
* Add code for checking statistic data in a single variable and collecting
* statistic data based on iStatsFrequency.
*
* - Add code to store statistic data in a single var
* - Collect statistic data once every iStatsFreqency transactions
* [from r1.10 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.1]
*
* Revision 1.10  1998/12/08  23:03:49  wenjian
* Add (or rename) Makefile for each platform (AIX and NT). Reorganize the
* files a little bit.
*
* - Define variable iStatsFrequency for AIX
* [from r1.9 by delta wenjian-23787-TPCC-integrate-code-for-AIX-and-NT, r1.1]
*
* Revision 1.9  1998/12/08  18:55:19  wenjian
* In pre_rpc, set headerP->stats to iStatsFrequency
* [from r1.8 by delta wenjian-23785-TPCC-pass-statsFrequency-from-client-to-server,r1.1]
*
* Revision 1.8  1998/12/07  20:04:12  wenjian
* Clean up
* [from r1.7 by delta wenjian-23742-TPCC-update-with-Raliegh-code,r1.2]
*
* Revision 1.7  1998/11/24  21:45:59  wenjian
* - Add #ifdef MULTIPLE_INTERFACE
* - Check if we need to collect statistics for response time
* - Do mutex_lock for terminal_context_init
* [from r1.6 by delta wenjian-23742-TPCC-update-with-Raliegh-code,r1.1]
*
* Revision 1.6  1998/11/09  16:59:38  wenjian
* In this revision, most of the changes are related to the directory of header
* files after directory reorganization. Other changes include adding or removing
* files to put them in the right directories. Makefiles are written for NT
* platform so that nmake is working on NT now. Need a top level Makefile for all
* the directories.
* [from r1.5 by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.2]
*
* Revision 1.5  1998/11/09  14:48:16  wenjian
* In an effort to make a new directory structure for TPCC, this delta
* creates two directories: tpcc/client and tpcc/server. All the files
* for this revision are copied from tpcc/sp-tpcc without any change.
* Further change may be needed for some files due to the change of
* the directory structure.
* [added by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.1]
*
* Revision 1.43  1998/11/06  16:10:55  wenjian
* - Minor change to reduce the print statement
* [from r1.42 by delta wenjian-23646-TPCC-clean-up-source-code, r1.1]
*
* Revision 1.42  1998/10/27  14:57:51  dongfeng
* Change enc_status to a data structure that has fields:
*   - Status code
*   - Line Number
*   - File Name
*   - Encina Error Code
*   - Error Msg
* Remove statusMsgs in web_tpcc.c
* [from r1.41 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.6]
*
* Revision 1.41  1998/10/26  14:41:34  dongfeng
* Add Init command in web client so when something bad happens during
* initialization web client sends back error information and allows
* reinitialization instead of killing IIS server.
*
* Define Macro CHK_STATUS instead of using #ifdef
* [from r1.40 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.3]
*
* Revision 1.40  1998/10/22  21:24:11  wenjian
* [merge of changes from 1.23 to 1.39 into 1.38]
*
* Revision 1.39  1998/10/22  19:18:34  dongfeng
* [from r1.37 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.2]
*
* Revision 1.37  1998/10/08  14:18:01  dongfeng
* Add codes for doing web-based tpcc.
* [from r1.23 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.1]
*
* Revision 1.38  1998/10/08  18:03:01  gerstl
* Changes to allow configurations where some servers only service
* specific transaction types. Split transaction interfaces by type.
* [from r1.36 by delta gerstl-23515-TPCC-allow-separate-online-transaction-interfaces,r1.1]
*
* Revision 1.36  1998/10/07  15:14:22  gerstl
* [merge of changes from 1.26 to 1.31 into 1.34]
*
* Revision 1.31  1998/09/04  19:17:56  wenjian
* Remove log_file_handle and related code.
* [from r1.29 by delta wenjian-23183-TPCC-sync-AIX-code-with-Austin,r1.5]
*
* Revision 1.29  1998/08/28  18:30:00  wenjian
* This delta sync the TPCC code with Austin.
*
* Remove UNCOND_EVENT in CALLTPCC, pre_rpc and post_rpc.
* [from r1.26 by delta wenjian-23183-TPCC-sync-AIX-code-with-Austin,r1.1]
*
* Revision 1.34  1998/09/26  10:56:26  oz
* - renamed thread_init and thread_done to clnt_thread_init and
*   clnt_thread_done respectively because of name conflicts on AIX4.3
* [from r1.26 by delta oz-23339-TPCC-update-for-NT, r1.2]
*
* Revision 1.26  1998/08/18  14:38:41  wenjian
* Remove adl.h from this file since it is not ported on NT. Use corresponding
* rpc protect levels and authz levels to replace ADL_.... macros defined in
* adl.h
* [from r1.23 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.4]
*
* Revision 1.23  1998/06/17  15:05:45  wenjian
* Somehow, read and write didn't work for socket on NT, although they
* are supposed to work. As a work-around way, use recv and send for
* NT in this revision. We may change them back if the problem is gone.
*
* Define SKIP_RPC and add code to do tests without calling DCE.
* This addition is for test purpose only.
* [from r1.22 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.2]
*
* Revision 1.22  1998/02/17  22:13:49  wenjian
* [merge of changes from 1.19 to 1.20 into 1.21]
*
* Revision 1.20  1998/02/17  16:04:42  oz
* - Split the login into two parts to allow for special logins
* - If the warehouse ID is 0, this is a special login to
*   query the client for status
*
* - Keep track of threads that have been initialized and also
*   threads that are done.
* [from r1.19 by delta oz-21864-TPCC-split-client-login-screen, r1.1]
*
* Revision 1.21  1998/02/17  22:07:03  wenjian
* Minor changes for NT
* [from r1.19 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.1]
*
* Revision 1.19  1998/01/26  20:37:35  oz
* - Remove all the code associated with explicit binding
*
* - Removed GET_SERVER_INDEX
* - Removed bindingType
* - Removed explicit binding from CALLTPCC
* - Removed calls to cancel_all_reservations and to init_handles
* [from r1.18 by delta oz-21697-TPCC-remove-explicit-binding-code, r1.1]
*
* Revision 1.18  1998/01/26  16:43:32  oz
* - Removed the code for collecting stats in the client
*   and dumping them before exit.
*
* - Removed pre_rpc_stats and post_rpc_stats
* - Removed code to write the stats out
* [from r1.17 by delta oz-21691-TPCC-remove-client-stats-code, r1.1]
*
* Revision 1.17  1998/01/26  16:19:23  oz
* - moved all the code pertaining to the background
*   thread to its own file and all the data structures
*   to client_utils.h
* [from r1.16 by delta oz-21689-TPCC-move-client-bg-thread-to-separate-file, r1.1]
*
* Revision 1.16  1998/01/26  15:33:32  oz
* - call impTPCCNOInfo to make sure there is a server out there
* [from r1.15 by delta oz-21671-TPCC-merge-online-transaction-interfaces, r1.2]
*
* Revision 1.15  1998/01/23  21:58:51  oz
* - In order to simplify the Encina TPCC code: Merge the four
*   online transactions into 1 interface
* - Moved all the scripts to a scripts subdirectory
* - Removed unused files
* [from r1.14 by delta oz-21671-TPCC-merge-online-transaction-interfaces, r1.1]
*
* Revision 1.14  1998/01/23  15:07:53  oz
* - Updated the SP TPCC directory to the latest files used
*   during the SP tpcc audit.
* [from r1.13 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
*
*
*
*/

#include "common/get_local_time.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdarg.h>
#include <time.h>
#if defined (solaris)
#include <dce/pthread.h>
#else   /* solaris */
#include <pthread.h>
#endif  /* solaris */
```

```
#include <tpm/mon/mon.h>
#include <utils/trace.h>
#include "common/delivery.h"
#ifdef MULTIPLE_INTERFACE
#include "common/neworder.h"
#include "common/payment.h"
#include "common/stocklevel.h"
#include "common/orderstatus.h"
#else
#include "common/tpcc_trans.h"
#endif
#include "common/utilities.h"
#include "client_utils.h"
#include "common/do_tpcc.h"
#include "client.h"
#include "encina_client.h"

#if 0
#define SKIP_RPC
#endif

extern void start_bg_debug_thread(void);
extern total_tran_count_t  *perfClntDataInit();

#define MAX_CONSECUTIVE_ERRORS 20

static void read_mon_environment(void);
static void client_trace(char *comp, int value, int add);
static void dump_pa_ring_buffer(trpc_handle_tpa_handle);

extern int warehouse_offset;

char Bndstring[512];

unsigned32     client_authnLevel;
unsigned32     client_authzSvc;

char *cellName;
int envRetrieval = 0;

static total_tran_count_t total_counts;  /* counts of transactions over
                                                                        * the entire test
                                                                        */


#ifndef WEB_TPCC_CLIENT
#undef CHK_STATUS
#define CHK_STATUS(status, val, a) if(status) {err_printf("Error: %d, %s\n",val,a); exit_program(status);}
MUTEX_T init_lock;
static int iStatsFrequency = 1;
#else
extern enc_status_t enc_status;
CRITICAL_SECTION    init_lock;
extern int iStatsFrequency;
#endif
int info_list_len= 0;
thread_info_t **info_list = NULL;  /*  List of all the thread info
                                                        *  structures.  This can be used
                                                        *  upon exit to cancel all the
                                                        *  reservations
                                                        */
total_tran_count_t  *pClientInfo=NULL;  /* keep stats for the client process */
static num_active_threads = 0;

#define NewOrder_code     NEWO_TRANS
#define Payment_code      PAYMENT_TRANS
#define OrderStatus_code  ORDER_STAT_TRANS
#define Delivery_code     DELIVERY_TRANS
#define StockLevel_code   STOCK_TRANS

extern int useSecurity;

#define INT_ENV_VALUE(var, default) \
 (var = getenv(#var) ? atoi(getenv(#var)) : default)

#define PRE_RPC_WORK(contextP, dataP, tran, sub_tran) \
      if (contextP != NULL) \
                              pre_rpc(contextP, &(dataP)->header, tran, sub_tran)
#define POST_RPC_WORK(contextP, dataP, tran) \
      if (contextP != NULL) \
                              post_rpc(contextP, &(dataP)->header, tran)
#define TIME_STR_P(infoP)  (&((infoP)->last_tran))

/*                  CALTPCC
*  Macro to sends 1 RPC and then handles any errors.
*
*  The macro takes the name of the RPC (e.g., NewOrder)
*  and makes the RPC by calling the appropriate function
*  (e.g., impTPCCNewOrder).
*/

/*
 switch(status) {
 case MON_FAILED : err_printf("mon_GetPaHandle failed\n"); break;
 case MON_INFO_UNAVAILABLE : err_printf("mon_GetPaHandle Info Unavailable\n");break;
 case MON_INVALID_PARM : err_printf("mon_GetPaHandle Invalid parameter\n"); break;
 case MON_SERVER_BUSY : err_printf("mon_GetPaHandle Server Busy\n"); break;
 case MON_SUCCESS : err_printf("mon_GetPaHandle Success paHandle=%lu\n",paHandle);break;
 }
*/

#ifdef SKIP_RPC
#define CALLTPCC(name,infoP,data,trpcStatusP)          \
{                                                      \
 struct timezone tz;                                                                                  \
 struct timespec timeP;                                                                               \
 char   tran_type[30];                                                                                \
 strcpy(tran_type,UTIL_STRING(name));                                                                 \
 timeP.tv_sec = 0;                                                                                    \
 timeP.tv_nsec = 190000000;                                                                           \
 if ( strcmp(tran_type,"NewOrder")==0 )                                                               \
    timeP.tv_nsec = 450000000;                                                                        \
 if ( strcmp(tran_type,"Payment")==0 )                                                                \
    timeP.tv_nsec = 90000000;                                                                         \
 pthread_delay_np(&timeP);                                                                            \
 gettimeofday(&TIME_STR_P(infoP)->send, &tz);                                                         \
}
#else
#define CALLTPCC(name,infoP,data,trpcStatusP)          \
{                                                      \
 struct timezone tz;                                                                                  \
 unsigned long status ;                                                                               \
 infoP->consecutive_errors = 0;                                                                       \
 retry:                                                                                               \
 if (infoP) gettimeofday(&TIME_STR_P(infoP)->send, &tz);                                              \
 UTIL_CONCAT(impTPCC,name)((encina_handle_t)Bndstring,data,trpcStatusP);                              \
 if (*(trpcStatusP)) {                                                                                \
  char msg[100];                                                                                      \
  if (infoP->consecutive_errors++ < MAX_CONSECUTIVE_ERRORS)                                           \
    goto retry_trans;                                                                                 \
  sprintf(msg, "TRPC error during impTPCC%s", UTIL_STRING(name));                                     \
  (data)->header.returncode = TRPC_ERROR;                                                             \
  encina_error_message(msg, *(trpcStatusP));                                                          \
 } else if (((data)->header.returncode != TPCC_SUCCESS) &&                                            \
          ((data)->header.returncode != INVALID_NEWO)) {                                              \
  char msg[100];                                                                                      \
  sprintf(msg, "App error during impTPCC%s: ", UTIL_STRING(name));                                    \
  encina_error_message(msg, (data)->header.returncode);                                               \
 }                                                                                                    \
}
#endif
#define CALLTPCCDEL(name,infoP,data,trpcStatusP)                                                      \
{                                                                                                     \
 struct timezone tz;                                                                                  \
 \
 infoP->consecutive_errors = 0;                                                                       \
 retry:                                                                                               \
 if (infoP) gettimeofday(&TIME_STR_P(infoP)->send, &tz);                                              \
 UTIL_CONCAT(impTPCC,name)(data,trpcStatusP);                                                         \
 if (*(trpcStatusP)) {                                                                                \
  if (infoP->consecutive_errors++ < MAX_CONSECUTIVE_ERRORS)                                           \
    goto retry_trans;                                                                                 \
  sprintf(msg, "TRPC error during impTPCC%s", UTIL_STRING(name));                                     \
  (data)->header.returncode = TRPC_ERROR;                                                             \
  encina_error_message(msg, *(trpcStatusP));                                                          \
 } else if (((data)->header.returncode != TPCC_SUCCESS) &&                                            \
          ((data)->header.returncode != INVALID_NEWO)) {                                              \
  char msg[100];                                                                                      \
  sprintf(msg, "App error during impTPCC%s: ", UTIL_STRING(name));                                    \
  encina_error_message(msg, (data)->header.returncode);                                               \
 }                                                                                                    \
}

/*
*  pre_rpc --   For debug purposes
*
*  Called before an RPC is made.
*  Set the state of the thread and keep track of the time the  RPC is sent.
*  This is used by the Background thread to report the state of the client.
*/
static void pre_rpc(thread_info_t *thread_infoP,
                    data_header *headerP,
                    int tran_type,
                    int sub_tran_type)
{
  tran_timing_t *curP;
  struct timezone tz;
  unsigned long status;

  curP = &thread_infoP->last_tran;
  curP->terminal = thread_infoP->thread_index;
  curP->tran = tran_type;
  curP->sub_tran = sub_tran_type;

  if (iStatsFrequency < 1) {
                    headerP->stats = 0;
  } else {
     int num;
     num = ++ (pClientInfo->tran[tran_type].num);
     headerP->stats = (num % iStatsFrequency==0) ? 1 : 0;
  }
  if (headerP->stats) { /* measure the time for RT */
                    gettimeofday(&curP->start, &tz);
                    headerP->start_time.sec = 0;
                    headerP->start_time.usec = 0;
                    headerP->end_time.sec = 0;
                    headerP->end_time.usec = 0;
```

```
#ifdef KEEP_TERMINAL_INFO
      set_client_debug_state((void *)thread_infoP, thread_state_sent, tran_type);
#endif
  }
}

/*
*  post_rpc
*
*  Called when the RPC returns from the server
*
*  Keeps track of the client response time and the server response time
*  as well as the state of the thread.  This is used by the background
*  debug thread to report the state of the client
*/
static void post_rpc(thread_info_t *thread_infoP,
                                        data_header *headerP,
                                        int tran_type)
{
  double time_diff_s,time_diff_c;
  tran_timing_t *curP;
  struct timezone tz;

  if (!thread_infoP) return;

  curP = &thread_infoP->last_tran;

  curP->server = headerP->dtype;  /*  The server sets this by convention */
  if (headerP->stats) {
                curP->srvr_start.tv_sec = headerP->start_time.sec;
                curP->srvr_start.tv_usec = headerP->start_time.usec;
                curP->srvr_done.tv_sec = headerP->end_time.sec;
                curP->srvr_done.tv_usec = headerP->end_time.usec;

     gettimeofday(&curP->end, &tz);
  }

#ifdef KEEP_TERMINAL_INFO
  /* Store the info for each terminal */
  thread_infoP->num_trans++;
  thread_infoP->tran[tran_type].num ++;
  if ((headerP->returncode == TPCC_SUCCESS) ||
      (headerP->returncode == INVALID_NEWO)) {
    thread_infoP->consecutive_errors = 0;
    curP->tran_failed = 0;
    if (headerP->returncode == INVALID_NEWO) {
      curP->sub_tran |= 0x100;
    }
  } else {
    thread_infoP->tran[tran_type].errs ++;
    thread_infoP->consecutive_errors++;
    curP->tran_failed = 1;
  }


  if (headerP->stats && tran_type <= MAX_TRAN_TYPE && tran_type > 0
      && !curP->tran_failed) {
     set_client_debug_state((void *)thread_infoP, thread_state_received, 0);

     /* update total server round trip response time */
     time_diff_s = time_diff_ms(&(curP->srvr_done),&(curP->srvr_start));
     thread_infoP->tran[tran_type].RTtotal[1] += time_diff_s;

     /* update total client round trip response time */
     time_diff_c = time_diff_ms(&(curP->end),&(curP->start));
     thread_infoP->tran[tran_type].RTtotal[0] += time_diff_c;

                /* update num for the number of trans which have RT measured */
     thread_infoP->tran[tran_type].RTcount ++;
  }

#else
  /* Store the info for each client.
   * Note: since we don't use mutex for performance reason, pClientInfo
         may not be accurate if more than one thread work on the same
         data at a same time. But this can reduce the overhead caused
         by scanning info_list for each terminal.
   */
  if ((headerP->returncode == TPCC_SUCCESS) ||
      (headerP->returncode == INVALID_NEWO)) {
    curP->tran_failed = 0;
    if (headerP->returncode == INVALID_NEWO) {
      curP->sub_tran |= 0x100;
    }
  } else {
    pClientInfo->tran[tran_type].errs ++;
    pClientInfo->errors ++;
    curP->tran_failed = 1;
  }
  if (headerP->stats && tran_type <= MAX_TRAN_TYPE && tran_type > 0
      && !curP->tran_failed) {
       /* update total server round trip response time */
       time_diff_s = time_diff_ms(&(curP->srvr_done),&(curP->srvr_start));
       pClientInfo->tran[tran_type].RTtotal[1] += time_diff_s;

       /* update total client round trip response time */
       time_diff_c = time_diff_ms(&(curP->end),&(curP->start));
       pClientInfo->tran[tran_type].RTtotal[0] += time_diff_c;
                /* update num for the number of trans which have RT measured */
       pClientInfo->tran[tran_type].RTcount ++;
  }
#endif
}

/*
* exit_program - restores original terminal attributes before leaving the
*                program.
*/

void exit_program( err )
 short int err;
{
  if ( err )
                fprintf(ERROUT , "exit_program: Error Code = %d\n", err );

  MUTEX_LOCK(&init_lock);
  /**  Cancel all the longterm reservations (if any)
   **               and write out the time-stamps
   **/

  if (info_list && (info_list_len > 0)) {
    int i;

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

                if (info_list[i] && info_list[i]->initialized){
                  info_list[i]->initialized = 0;
                }
    }
  }

  MUTEX_UNLOCK(&init_lock);

  if (logtpcc) {
    fclose(logtpcc);
  } else {
    if (logtpcc = fopen(log_file_name,"w")) {
                fprintf(logtpcc, "ERROR: Client exiting before SYNC with error %d\n",
                                err);
                fclose(logtpcc);
    }
  }

  mon_ExitClient( err );

#ifndef WEB_TPCC_CLIENT
  exit( err );
#endif
}

/*
*  clnt_thread_init
*
*  This function must be called by each work thread
*  It returns a pointer to a context that must be passed
*  on calls back to this module.
*  There is 1 threadInfo entry in an array for each executor thread.
*  When an executor thread is started the first thing it does is call
*  this clnt_thread_init function.  This function creates a context for the
*  thread and if longterm reservations are used this function
*  initializes the pa handle.
*/
void *clnt_thread_init(void)
{
  int thread_index;
  struct timezone tz;
  thread_info_t *thread_infoP;

  if (iStatsFrequency < 1)
                return(NULL);

  thread_infoP = (thread_info_t *)calloc(1, sizeof(thread_info_t));

  thread_infoP->descr.state = thread_state_init;
  gettimeofday(&thread_infoP->descr.init,&tz);
  thread_infoP->initialized = 1;

  MUTEX_LOCK(&init_lock);
  thread_index = info_list_len++;
  thread_infoP->thread_index = thread_index;
  thread_infoP->thread_id = get_thread_id();

  num_active_threads++;
  info_list =
                (thread_info_t **)realloc((void *)info_list,
                                                                sizeof(thread_info_t *) * info_list_len);

  info_list[thread_index] = thread_infoP;

  MUTEX_UNLOCK(&init_lock);

  if (num_active_threads % 200 == 0)
     err_printf("Thread %d Initialized (currently %d are active).\n",
                        thread_index, num_active_threads);


  return(thread_infoP);
```

```
}

/*
*  clnt_thread_done
*
*  Called before a thread exits.
*  Perform some cleanup.
*
*/
void clnt_thread_done(contextP)
 void *contextP;
{
  int all_done = 0;
  int j;
  thread_info_t *infoP = (thread_info_t *)contextP;

  if (!infoP) return;


  MUTEX_LOCK(&init_lock);

  num_active_threads--;

#if 0
  err_printf("> thread_done, %d active\n", num_active_threads);
#endif

  set_client_debug_state((void *)infoP, thread_state_done, 0);
  infoP->done = 1;

  if (num_active_threads == 0) {
    all_done = 1;
  }

  if (info_list[infoP->thread_index]!= infoP) {
    fprintf(ERROUT, "Strange error: expected to find %d in info_list[%d] and found %d instead\n",
                        infoP, infoP->thread_index,
                        info_list[infoP->thread_index]);
  }

  MUTEX_UNLOCK(&init_lock);
  if (all_done) {
                int i;
                thread_info_t **curP;
#if 0
                fprintf(ERROUT, "All Done - exiting\n");
#endif
                MUTEX_LOCK(&init_lock);
                for (i=0, curP=info_list; i<info_list_len;i++, curP++) {
                    free(*curP);
                }
                free(info_list);
                info_list = NULL;
                info_list_len = 0;
                MUTEX_UNLOCK(&init_lock);
#if 0
                exit(0);
#endif
  }
}

/*
*  The following send_*** functions are called from the screen
*  module after the transaction data is received in order to
*  send the data to the server for processing.
*
*/

/*
*  send_new_order
*               Send a new order request to the server
*/
void send_new_order(contextP, dataP)
 void *contextP;
 newOrder_data_t *dataP;
{
  thread_info_t *thread_context = (thread_info_t *)contextP;
  trpc_status_t trpcStatus;

  DPRINT(("New Order, w_id %d, %d orders\n", dataP->w_id, dataP->o_ol_cnt));
  PRE_RPC_WORK(thread_context, dataP, NEWO_TRANS, dataP->o_all_local == 0);
  CALLTPCC(NewOrder,thread_context,dataP,&trpcStatus)
  POST_RPC_WORK(thread_context, dataP, NEWO_TRANS);
}

/*
*  send_payment
*               Send a payment request to the server
*/
void send_payment(contextP, dataP)
 void *contextP;
 payment_data_t *dataP;
{
  trpc_status_t trpcStatus;
  thread_info_t *thread_context = (thread_info_t *)contextP;

  PRE_RPC_WORK(thread_context, dataP, PAYMENT_TRANS,
                                dataP->w_id != dataP->c_w_id);
  CALLTPCC(Payment,thread_context,dataP,&trpcStatus);
  POST_RPC_WORK(thread_context, dataP, PAYMENT_TRANS);
}

/*
*  send_order_status
*               Send a order status request to the server
*/
void send_order_status(contextP, dataP)
 void *contextP;
 orderStatus_data_t *dataP;
{
  trpc_status_t trpcStatus;
  thread_info_t *thread_context = (thread_info_t *)contextP;

  PRE_RPC_WORK(thread_context, dataP, ORDER_STAT_TRANS, 0);
  CALLTPCC(OrderStatus,thread_context,dataP,&trpcStatus);
  POST_RPC_WORK(thread_context, dataP, ORDER_STAT_TRANS);
}

/*
*  send_delivery
*               Send a delivery request to the server
*/
void send_delivery(contextP, dataP)
 void *contextP;
 delivery_data_t *dataP;
{
  trpc_status_t trpcStatus;
  thread_info_t *thread_context = (thread_info_t *)contextP;

  PRE_RPC_WORK(thread_context, dataP, DELIVERY_TRANS, 0);
  CALLTPCCDEL(Delivery,thread_context,dataP,&trpcStatus);
  POST_RPC_WORK(thread_context, dataP, DELIVERY_TRANS);
}

/*
*  send_stock_level
*               Send a stock level request to the server
*/
void send_stock_level(contextP, dataP)
 void *contextP;
 stockLevel_data_t *dataP;
{
  trpc_status_t trpcStatus;
  thread_info_t *thread_context = (thread_info_t *)contextP;

  PRE_RPC_WORK(thread_context, dataP, STOCK_TRANS, 0);
  CALLTPCC(StockLevel,thread_context,dataP,&trpcStatus);
  POST_RPC_WORK(thread_context, dataP, STOCK_TRANS);
}

int too_many_errors(contextP)
 void *contextP;
{
 thread_info_t *thread_context = (thread_info_t *)contextP;

 return (thread_context->consecutive_errors > MAX_CONSECUTIVE_ERRORS);
}

/*
*  Enroll the client:
*               Perform the needed initialization
*/
void init_encina_client(user_id)
 int user_id;
{
  int i;
  mon_status_t                  monStatus;
  char *env_str;
  char serverName[48];
  struct timezone tz;
  struct timeval a_time;
  unsigned long status ;
  FILE  *rcFile;

#ifdef WIN32
  get_time_init();
  pClientInfo = perfClntDataInit();
#endif
  if (pClientInfo == NULL)
                pClientInfo = malloc(sizeof(total_tran_count_t));
  memset(pClientInfo, 0, sizeof(total_tran_count_t));

  read_mon_environment();

  if(!cellName)
                CHK_STATUS(30, CELL_NAME_UNAVAILABLE,
              "ENCINA_TPM_CELL is not set!");

  MUTEX_INIT(&init_lock);

  info_list = NULL;
  info_list_len = 0;

#ifndef WEB_TPCC_CLIENT
  /* initialize iStatsFrequency */
  iStatsFrequency = 1;
  rcFile = fopen("~/.tpccrc", "r");
  if (rcFile!=NULL) {
```

```
    char buf[100];
                int num = 1;
                while (1) { /* read the whole rcFile */
      num = fscanf(rcFile,"%s",buf);
      if (num<=0) break;
                  if (strcasecmp(buf,"StatsFrequency")==0) {
                      fscanf(rcFile,"%d",&iStatsFrequency);
        break;
      }
    }
  }
  err_printf("iStatsFrequency=%d\n",iStatsFrequency);
#endif

  gettimeofday(&a_time,&tz);
#ifdef WIN32
  srand(a_time.tv_sec ^ a_time.tv_usec);
#else
  srand48(a_time.tv_sec ^ a_time.tv_usec);
#endif
}


/*
*  Enroll the client:
*               get the necessary handles.
*/
void enroll_client(user_id)
 int user_id;
{
  mon_status_t                  monStatus;
  char *env_str;
  static  char  *clientName="tpcc_client";
  unsigned long status ;
  static  int   client_enrolled= 0;
  FILE *bindmap;
  rpc_if_handle_t ifSpec;
  int uport;
  char server[512];
  char *tmp;
  char line[512];
  int a,i=0,j;
  char **serverList;
  unsigned long serverCount;
  extern int user_port;

  MUTEX_LOCK(&init_lock);
  if (client_enrolled) {
                MUTEX_UNLOCK(&init_lock);
                return;
  }
  if (useSecurity) {
                client_authnLevel= rpc_c_protect_level_connect;
                client_authzSvc = rpc_c_authz_dce;
  } else {
                client_authnLevel= rpc_c_protect_level_none;
                client_authzSvc = rpc_c_authz_none;
  }

  if (envRetrieval == 0) {
     ENCINA_CALL_RC("mon_RetrieveEnable",mon_RetrieveEnable(FALSE),status);
     CHK_STATUS(status, MON_RETRIEVEENABLE_FAILED,
            "mon_RetrieveEnable failed");
  }

  DPRINT(("Cell name: %s\n", cellName));

  ENCINA_CALL_RC("mon_InitClient",mon_InitClient(clientName,cellName),
status);
  CHK_STATUS(status, MON_INITCLIENT_FAILED,
          "mon_InitClient failed");

  DPRINT(("mon_SecuritySetDefaults-> authn %d, authz %d\n",
                  client_authnLevel,client_authzSvc));
  ENCINA_CALL_RC("mon_SecuritySetDefaults",
                                mon_SecuritySetDefaults(client_authnLevel,client_authzSvc),
          status);
  CHK_STATUS(status, MON_SECURITYSET_FAILED,
          "mon_SecuritySetDefaults failed");

  ENCINA_CALL_RC("mon_SetHandleCacheRefreshInterval",
                                mon_SetHandleCacheRefreshInterval(300),status);
  CHK_STATUS(status, MON_SETREFRESHINTERVAL_FAILED,
          "mon_SetHandleCacheRefreshInterval failed");


  bindmap = fopen("/u/encina/.tpccbindrc","r");

  if (bindmap) {
    err_printf("Open /home/encina/.tpccbindrc for reading\n");
    do {
                int found=0;
      if (fgets(line, 512, bindmap) == NULL)
                break;

          sscanf(line,"%d %s", &uport, server);

          if (uport == user_port) {
              /* Get list of servers in the Cell frpm CDS*/
              if ((monStatus = mon_GetMondList(neworder_v1_0_c_ifspec,&serverCount, &serverList)) !=
MON_SUCCESS) {
                switch(monStatus) {
                                case MON_INFO_UNAVAILABLE: CHK_STATUS(status,
MON_INFO_UNAVAILABLE, "mon_GetMondList failed, info unavailable."); break;
                                case MON_INVALID_PARM: CHK_STATUS(status,
MON_INVALID_PARM, "mon_GetMondList failed, invalid parameter."); break;
                 }
              }

              for (i=0; i < serverCount; i++) {
                                a = strlen(serverList[i]);
                tmp = serverList[i] + (a - strlen(server));
                if (strstr(tmp,server)) {
                                 strcpy(Bndstring,"server:");
                    strcat(Bndstring,strrchr(serverList[i],'/')+1);
                                err_printf("Bind String: %s\n",Bndstring);
                                mon_Free(serverList);
                                found = 1;
                                break;
                  }
                  if (i == serverCount) {
                    mon_Free(serverList);
                    err_printf("Port %d, Cannot find server.\n",uport);
                    exit_program(43);
                  }
                 }
                 if (found)
                   break;
                }
          } while (1);
    fclose(bindmap);
    } else
    err_printf("Cannot open /home/encina/.tpccbindrc for reading\n");


  {
     dbInfo_data_t data;
     trpc_status_t trpcStatus;
     /* Get DB Info -- currently id does not do anything
       but it will tell us if there is a server out there.
       Better to know instead of when all the terminals
       are up and ready
      */

     impTPCCNOInfo((encina_handle_t)Bndstring,&data,&trpcStatus);

                err_printf("serverIdNumber=%d\n",data.server_id);
     if (trpcStatus) {
        char msg[100];
        sprintf(msg, "TRPC error during db info at init.");
        encina_error_message(msg,trpcStatus);
        CHK_STATUS(33,NOINFO_TRPC_ERROR,
                "TRPC error during db info at init");
     }
  }

  /* Start bg_thread for debug purpose and performance tuning.
   * In the final test, we do not start it in order to get the
   * best performance.
   * On NT, bg_thread may use lots of CPU. But we need to verify it.
   */
  if (1)
                start_bg_debug_thread();

  client_enrolled= 1;
  MUTEX_UNLOCK(&init_lock);

}

/*-----------------------------------------------------------------*/
/*       Read environment paramaters                               */
/*-----------------------------------------------------------------*/
static void read_mon_environment()
{
  char *env_str;


  cellName= getenv("ENCINA_TPM_CELL");
  CHECK_ENVIRON(cellName,"ENCINA_TPM_CELL");

  if (env_str = getenv("TPCC_ENV_RETRIEVE")) {
                envRetrieval= atoi(env_str);
  }
}

/*
*  dump_pa_ring_buffer()-- For Debugging --
*      Dump the ring buffer in the PA we are talking to
*      Only works if we are using long term reservation
*/
static void dump_pa_ring_buffer(pa_handle)
 trpc_handle_t pa_handle;
{
 err_printf("Dumping Ring Buffer of server\n");
 admin_trace_DumpRingBuffer((handle_t)pa_handle,"stderr");
}
```

```
/* terminal_context_init:
*    The same function as thread_init in the thread-pool version.
*
*  This function must be called by each terminal when using
*  thread pool. After a terminal is logged on, the first thing
*  it does is to call this function.
*  This function creates a context for the terminal.
*  It returns a pointer to a context that must be passed
*  on calls back to this module.
*/
void *terminal_context_init(int fdIn)
{
  int thread_index;
  struct timezone tz;
  thread_info_t *thread_infoP;

  if (iStatsFrequency < 1)
                return(NULL);

  thread_infoP = (thread_info_t *)calloc(1, sizeof(thread_info_t));

  thread_infoP->descr.state = thread_state_init;
  gettimeofday(&thread_infoP->descr.init,&tz);
  thread_infoP->initialized= 1;

  MUTEX_LOCK(&init_lock);
  thread_index = info_list_len++;
  thread_infoP->thread_index = thread_index;
  thread_infoP->thread_id = fdIn;

  num_active_threads++;
  info_list =
     (thread_info_t **)realloc((void *)info_list, \
                        sizeof(thread_info_t *) *\
                         info_list_len);
  info_list[thread_index]= thread_infoP;

  MUTEX_UNLOCK(&init_lock);

  if (num_active_threads % 200 == 0)
     err_printf("Terminal %d Initialized (currently %d are\
active).\n",\
             thread_index, num_active_threads);


   return(thread_infoP);
}
```

neworder.tacf

```
/*
 * Copyright (C) 1991, 1990 Transarc Corporation
 * All Rights Reserved
 */
/*
 * neworder.tacf -- attribute configuration file for tpcc server.
 * used for transparent binding
 *
 * $Revision: 1.1 $
 * $Date: 1998/11/06 21:10:13 $
 * $Log:          neworder.tacf,v $
 *
 * $TALog: neworder.tacf,v $
 * Revision 1.1  1998/11/06  21:10:13  dongfeng
 * - Move all files common to client and server to tpcc/common
 *   directory
 * [added by delta dongfeng-23677-TPCC-new-directory-structures, r1.1]
 *
 * Revision 1.2  1998/10/08  18:03:01  gerstl
 * Changes to allow configurations where some servers only service
 * specific transaction types. Split transaction interfaces by type.
 * [added by delta gerstl-23515-TPCC-allow-separate-online-transaction-interfaces,r1.1]
 *
 *
 *
 */
[explicit_handle encina_handle_t]
interface neworder
{
}
```

neworder.tidl

```
/*
* id: $id: $
*
* component_name: encina benchmarks
*
* the following functions list may not be complete.
* functions defined by/via macros may not be included.
*
* functions:
*    <fill_me_in>
*
* origins: transarc corp.
*
* (c) copyright transarc corp. 1995, 1993
* all rights reserved
* licensed materials - property of transarc
*
* us government users restricted rights - use, duplication or
* disclosure restricted by gsa adp schedule contract with transarc corp
*/
/*
* history
* $talog: $
*/

/*
* neworder.tidl -- interface definition file fortpccserver.
*
* $revision: 1.0 $
* $date: 1995/10/20 21:55:05 $
* $log:          tpcc.tidl,v $
*/

[
   uuid(f7065094-5e04-11d2-b351-9e621208aa77),
   version(1.0)
]
interface neworder
{
import "tpm/mon/mon_handle.idl";
import "tpcc_type.idl";

[nontransactional] void impTPCCNewOrder(
                                                [in,out] newOrder_data_t *dataP,
                                                [out] trpc_status_t  * trpcStatus);
[nontransactional] void
      impTPCCNOInfo(
                                        [out] dbInfo_data_t *dataP,
                [out] trpc_status_t  * trpcStatus);
}
```

orderstatus.tacf

```
/*
 * Copyright (C) 1991, 1990 Transarc Corporation
 * All Rights Reserved
 */
/*
 * orderstatus.tacf -- attribute configuration file for tpcc server.
 * used for transparent binding
 *
 * $Revision: 1.1 $
 * $Date: 1998/11/06 21:10:14 $
 * $Log:          $
 *
 * $TALog: orderstatus.tacf,v $
 * Revision 1.1  1998/11/06  21:10:14  dongfeng
 * - Move all files common to client and server to tpcc/common
 *   directory
 * [added by delta dongfeng-23677-TPCC-new-directory-structures, r1.1]
 *
 * Revision 1.2  1998/10/08  18:03:02  gerstl
 * Changes to allow configurations where some servers only service
 * specific transaction types. Split transaction interfaces by type.
 * [added by delta gerstl-23515-TPCC-allow-separate-online-transaction-interfaces,r1.1]
 *
 *
 */

[explicit_handle encina_handle_t]
interface orderstatus
{
}
```

orderstatus.tidl

```
/*
* id: $id: $
*
* component_name: encina benchmarks
*
* the following functions list may not be complete.
* functions defined by/via macros may not be included.
*
* functions:
```

```
*    <fill_me_in>
*
* origins: transarc corp.
*
* (c) copyright transarc corp. 1995, 1993
* all rights reserved
* licensed materials - property of transarc
*
* us government users restricted rights - use, duplication or
* disclosure restricted by gsa adp schedule contract with transarc corp
*/
/*
* history
* $talog: $
*/

/*
* orderstatus.tidl -- interface definition file fortpccserver.
*
* $revision: 1.0 $
* $date: 1995/10/20 21:55:05 $
* $log:            tpcc.tidl,v $
*/
[
  uuid(06287200-5e05-11d2-8984-9e621208aa77),
  version(1.0)
]
interface orderstatus
{
import "tpm/mon/mon_handle.idl";
import "tpcc_type.idl";

[nontransactional] void    impTPCCOrderStatus(
                                        [in,out] orderStatus_data_t *dataP,
                                        [out] trpc_status_t  * trpcStatus);
}
```

payment.tacf

```
/*
* Copyright (C) 1991, 1990 Transarc Corporation
* All Rights Reserved
*/
/*
* payment.tacf -- attribute configuration file for tpcc server.
* used for transparent binding
*
* $Revision: 1.1 $
*
*/
[explicit_handle encina_handle_t]
interface payment
{
}
```

payment.tidl

```
/*
* payment.tidl -- interface definition file fortpccserver.
*
* $revision: 1.0 $
* $date: 1995/10/20 21:55:05 $
* $log:            tpcc.tidl,v $
*/
[
  uuid(1341a902-5e05-11d2-bb70-9e621208aa77),
  version(1.0)
]
interface payment
{
import "tpm/mon/mon_handle.idl";
import "tpcc_type.idl";

[nontransactional] void    impTPCCPayment(
                                        [in,out] payment_data_t *dataP,
                                        [out] trpc_status_t  * trpcStatus);
}
```

screen.C

```
/* (C)1997 IBM Corporation */
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <ctype.h>
#include <string.h>
#include <math.h>

#include "screen.h"
#include "format.h"
#include "encina.h"

#define USE_INSULTS
#define LOCAL_SESSION_DATA

extern "C" err_printf(...);

extern char const * const blanks;
extern char const * const underscores;
extern char const * const backspaces;

static int clear_eos(InOut *ioP);
static int clear_eos(char *buf);
static int string_empty(char const *text);
static int pos_zero(int const *val);
static int pos_nonzeros(int const **val);

/*****************************************************************************
Screen
*****************************************************************************/
int Screen::reset() {
    has_data=0;
    pos=0;
    if (dataptr) memset(dataptr, 0, data_len);
    for (int i = 0; fields && fields[i] != NULL; i++) {
                fields[i]->reset();
    }
    return 0;
};
int Screen::present_empty_fields(){
    if (empty_fields)
                threadP->write(empty_fields,empty_fields_len);
//    threadP->write(end_str, end_str_len);
    return 0;
}
int Screen::present() {
    threadP->write(screen, screen_len);
    threadP->write(session_data, session_data_len);
    if (has_data) {
                for (int i = 0; fields[i] != NULL; i++) {
                    fields[i]->display_field(1);
                }
//              threadP->write(end_str, end_str_len);
    } else {
                present_empty_fields();
    }
    return 0;
};
int Screen::user_input() {
    int key;
    has_data = 1;
    fields[pos]->start_position();
    threadP->flush();
//    threadP->mark();
    key = fields[pos]->get_field(0);
    do {
                switch (key) {
                case EOF:
                    return 0;
                    break;
                case Field::NEXT_FIELD:
                    if (fields[++pos] == NULL) {
                                pos = 0;
                    }
                    break;
                case Field::PREV_FIELD:
                    if (--pos < 0) {
                                while (fields[++pos] != NULL);
                                pos--;
                    }
                    break;
                case Field::REDISPLAY:
                    present();
                    break;
                case Field::ABORT:
//                  position(1, 2);
                    threadP->write(end_str, end_str_len);
                    return 0;
                case Field::ENTER:
                    if (validate()) {
//                              threadP->pop();
                                return 1;
                    }
                    break;
                }
                key = fields[pos]->get_field();
    } while (1);
    return 0;
}
Screen::~Screen() {
    if (fields != NULL) {
                for (int lpos = 0; fields[lpos] != NULL; lpos++) {
                    delete fields[lpos];
                }
                delete [] fields;
    }
```

```
    fields=NULL;
}

int Screen::display_status(int status) {
    position(threadP, status_x, status_y);
    threadP->write("Execution Status: ");
    if (status == TRAN_OK) {
                threadP->write("Transaction Committed");
    } else if (status == INVALID_ITEM) {
                threadP->write("Item number is not valid");
    } else {
                threadP->write("ERROR: Rollback -- ");
//              threadP->write("Rollback -- ");
                char buf[6];
                format_int(buf, 6, status);
                threadP->write(buf, 5);
    }
    return 0;
}

int Screen::handle() {
    threadP->debug("%s - reset\n", tran_type);
    reset();

    threadP->debug("%s - present\n", tran_type);
    threadP->hold();
    present();
    threadP->write(TRIGGER, 1);
    threadP->debug("%s - user_input\n", tran_type);
    if (!user_input()) {
                threadP->write(end_str, end_str_len);
                threadP->write(TRIGGER, 1);
                return -1;
    }
    threadP->flush();
    threadP->hold();
    threadP->debug("%s - process\n", tran_type);
    if (process()) {
                threadP->write(end_str, end_str_len);
                threadP->write(TRIGGER, 1);
                return -1;
    }
    threadP->debug("%s - respond\n", tran_type);
    respond();
    //    position(threadP, 1, 2);
    threadP->write(end_str, end_str_len);
    threadP->write(TRIGGER, 1);
    threadP->flush();
    return 0;
}

/*****************************************************************************
NewOrder
*****************************************************************************/
int NewOrder::reset() {
    Screen::reset();
    pos=start_field;
    memset(dataptr, 0, sizeof(*data));
    return 0;
};
NewOrder::NewOrder(User_data *udP, Thread_data *threadP) : Screen(udP, threadP) {
    tran_type = NEWORDER_SERVICE;
    dataptr = data = new NewOrder_data;
    data_len = sizeof(NewOrder_data);


    status_x = 1;
    status_y = 24;

    screen          = static_screen;
    empty_fields    = static_empty_fields;
#ifdef LOCAL_SESSION_DATA
    session_data = new char[static_session_data_len+1];
    sprintf(session_data, "%s%5d", POS(12,4), user_dataP->warehouse);
#else
    session_data    = static_session_data;
    sprintf(session_data, "%s%5d", POS(12,4), user_dataP->warehouse);
#endif
    screen_len      = static_screen_len;
    empty_fields_len= static_empty_fields_len;
    session_data_len= static_session_data_len;

    int lpos = 0;
    fields = new Field*[2+MAX_ITEMS*3+1];
    for (int i = 0; i < MAX_ITEMS; i++) {
                fields[lpos++]= genfield(threadP,  3, 9+i, 5, &data->item[i].s_OL_SUPPLY_W_ID);
                fields[lpos++]= genfield(threadP, 10, 9+i, 6, &data->item[i].s_OL_I_ID);
                fields[lpos++]= genfield(threadP, 45, 9+i, 2, &data->item[i].s_OL_QUANTITY);
#if defined(USE_SMART_FIELDS)
                if (i > 0) {
                    int **tmp = new int *[4];
                    tmp[0] = &fields[lpos-6]->pos;
                    tmp[1] = &fields[lpos-5]->pos;
                    tmp[2] = &fields[lpos-4]->pos;
                    tmp[3] = NULL;
                    fields[pos-3]->ok_func= (int(*)(void*))pos_nonzeros;
                    fields[pos-3]->ok_data = tmp;
                    fields[pos-2]->ok_func= (int(*)(void*))pos_nonzeros;
                    fields[pos-2]->ok_data = tmp;
                    fields[pos-1]->ok_func= (int(*)(void*))pos_nonzeros;
                    fields[pos-1]->ok_data = tmp;
                }
#endif
    }
    start_field = lpos;
    fields[lpos++]= genfield(threadP, 29, 4, 4, &data->s_D_ID); /* District */
    fields[lpos++]= genfield(threadP, 12, 5, 4, &data->s_C_ID); /* Customer */
    fields[lpos++]= NULL;
    reset();
};

int NewOrder::validate() {
    if (!fields[start_field]->pos){
                pos=start_field;
                message(threadP, "District ID is a required field");
                return 0;
    }
    if (!fields[start_field+1]->pos){
                pos=start_field+1;
                message(threadP, "Customer ID is a required field");
                return 0;
    }

    int last=-1;
    data->s_O_OL_CNT = 0;
    data->s_all_local = 1;
    data->s_W_ID = user_dataP->warehouse;
    for (int i = 0; i < MAX_ITEMS*3; i+=3 ) {
                if (fields[i]->pos|| fields[i+1]->pos|| fields[i+2]->pos){
                    if (!fields[i]->pos){
                                pos=i;
#if defined(USE_INSULTS)
                                message(threadP, "Yeah, I think this is a bogus field too.");
#else
                                message(threadP, "Warehouse ID is a required field");
#endif
                                return 0;
                    }
                    if (!fields[i+1]->pos){
                                pos=i+1;
#if defined(USE_INSULTS)
                                message(threadP, "Umm, WHAT did you want?");
#else
                                message(threadP, "Item ID is a required field");
#endif
                                return 0;
                    }
                    if (data->item[i/3].s_OL_QUANTITY <= 0) {
                                pos=i+2;
#if defined(USE_INSULTS)
                                message(threadP, "So something plus nothing is...");
#else
                                message(threadP, "Please enter a quantity greater than 0");
#endif
                                return 0;
                    }
                    if (data->item[i/3].s_OL_SUPPLY_W_ID != data->s_W_ID) {
                                data->s_all_local=0;
                    }
                    data->s_O_OL_CNT++;
                } else if (last < 0) {
                    last = i;
                }
    }
    if (data->s_O_OL_CNT <= 0) {
                pos=0;
#if defined(USE_INSULTS)
                message(threadP, "It's kind of pointless without ordering something isn't it?");
#else
                message(threadP, "Please enter an item to order");
#endif
                return 0;

    }
    // Compress the order lines:  some of them may be empty
    int ind;
    for (i=0, ind=0; ind<data->s_O_OL_CNT ; i++ ) {
        if (fields[i*3]->pos) {
            if (i > ind) {
                data->item[ind] = data->item[i];
            }
            ind ++;
        }
    }
    if (i > ind) {
                int j;
                for (j=ind; j<i; j++) {
                    /*  At least one empty line was skipped */
                    data->item[j].s_OL_SUPPLY_W_ID = 0;
                    data->item[j].s_OL_I_ID = 0;
                    data->item[j].s_OL_QUANTITY = 0;
                }
    }

    return 1;
}

int NewOrder::respond() {
```

```
  int i;
  double amount, total_amount, cost;
  char buf[32];
  position(threadP,  1, 9); clear_eos(threadP);
  position(threadP, 25, 5); threadP->write(data->s_C_LAST);
  position(threadP, 52, 5); threadP->write(data->s_C_CREDIT);
  position(threadP, 15, 6); format_int(buf,10, data->s_O_ID); threadP->write(buf, 9);

  display_status(data->s_transtatus);
  if (data->s_transtatus != TRAN_OK) {
                return -1;
  }

  position(threadP, 25, 5); threadP->write( data->s_C_LAST);
  position(threadP, 52, 5); threadP->write( data->s_C_CREDIT);
  position(threadP, 15, 6); format_int( buf, 10, data->s_O_ID); threadP->write(buf, 9);
  position(threadP, 48, 6); format_int( buf, 3, data->s_O_OL_CNT); threadP->write(buf, 2);
  position(threadP, 61, 4); format_date(buf, 20, data->s_O_ENTRY_D); threadP->write(buf, 19);
  position(threadP, 64, 5); format_float(buf, 6, 2, data->s_C_DISCOUNT/100); threadP->write(buf, 5);
  position(threadP, 59, 6); format_float(buf, 6, 2, data->s_W_TAX/100); threadP->write(buf, 5);
  position(threadP, 74, 6); format_float(buf, 6, 2, data->s_D_TAX/100); threadP->write(buf, 5);
  total_amount = 0;
  for (i=0; i < data->s_O_OL_CNT; i++) {
                position(threadP,  3, 9+i); format_int(buf, 6, data->item[i].s_OL_SUPPLY_W_ID);
threadP->write( buf, 5 );
                position(threadP, 10, 9+i); format_int(buf, 7, data->item[i].s_OL_I_ID); threadP->write(
buf, 6 );
                position(threadP, 19, 9+i); threadP->write( data->item[i].s_I_NAME);
                position(threadP, 45, 9+i); format_int(buf, 3, data->item[i].s_OL_QUANTITY);
threadP->write(buf, 2);
                position(threadP, 51, 9+i); format_int(buf, 4, data->item[i].s_S_QUANTITY);
threadP->write(buf, 3);
                position(threadP, 58, 9+i); threadP->write(&data->item[i].s_brand_generic, 1);
                position(threadP, 62, 9+i); format_money(buf, 8, data->item[i].s_I_PRICE);
threadP->write(buf, 7);
                position(threadP, 71, 9+i); format_money(buf, 10, data->item[i].s_OL_AMOUNT);
threadP->write(buf, 9);
  }
  /* Clear the screen of any empty input fields*/
  position(threadP, 63, 24); threadP->write( "Total:");
  position(threadP, 70, 24); format_money( buf, 10, data->s_total_amount ); threadP->write( buf, 9 );

  return 0;
}

/*****************************************************************************
Payment
*****************************************************************************/
Payment::Payment(User_data *udP, Thread_data *threadP)  : Screen(udP, threadP) {
  tran_type = PAYMENT_SERVICE;
  dataptr = data = new Payment_data;
  data_len = sizeof(Payment_data);

  int lpos = 0;
  screen          = static_screen;
  empty_fields    = static_empty_fields;
#ifdef LOCAL_SESSION_DATA
  session_data = new char[static_session_data_len+1];
  sprintf(session_data, "%s%5d", POS(12,6), user_dataP->warehouse);
#else
  session_data    = static_session_data;
  sprintf(session_data, "%s%5d", POS(12,6), user_dataP->warehouse);
#endif
  screen_len      = static_screen_len;
  empty_fields_len= static_empty_fields_len;
  session_data_len= static_session_data_len;

  fields = new Field*[7];
  fields[lpos++]= genfield(threadP, 52,  6,  2, &data->s_D_ID); /* District */
  fields[lpos++]= genfield(threadP, 11, 11,  4, &data->s_C_ID); /* Customer # */
  fields[lpos++]= genfield(threadP, 29, 12, 16, (char *)data->s_C_LAST); /* Name */
  fields[lpos++]= genfield(threadP, 33, 11,  5, &data->s_C_W_ID); /* Cust-Warehouse */
  fields[lpos++]= genfield(threadP, 54, 11,  2, &data->s_C_D_ID); /* Cust-District */
  fields[lpos++]= genfield(threadP, 23, 17,  8, &data->s_H_AMOUNT); /* Amount Paid */
  fields[lpos++]= NULL;
#if defined(USE_SMART_FIELDS)
  fields[1]->ok_func= (int(*)(void*))pos_zero;
  fields[1]->ok_data= &fields[2]->pos;
  fields[2]->ok_func= (int(*)(void*))pos_zero;
  fields[2]->ok_data= &fields[1]->pos;
#endif
};

int Payment::validate() {
  if (!fields[0]->pos) {
                pos=0;
                message(threadP, "District ID is a required field");
                return 0;
  }
  if (fields[1]->pos) {
#if defined(USE_BYNAME)
                data->s_byname= 0;
#endif
  } else if (fields[2]->pos) {
#if defined(USE_BYNAME)
                data->s_byname= 1;
#endif
  } else {
                pos=1;
                message(threadP, "Customer ID or Name is required");
                return 0;
  }

  if (!fields[3]->pos) {
                pos=3;
                message(threadP, "Customer Warehouse is a required field");
                return 0;
  }

  if (!fields[4]->pos) {
                pos=4;
                message(threadP, "Customer District is a required field");
                return 0;
  }

  if (data->s_H_AMOUNT <= 0) {
                pos=5;
                message(threadP, "Enter a positive amount");
                return 0;
  }

  data->s_W_ID = user_dataP->warehouse;

  return 1;
}

int Payment::respond() {
  if (data->s_transtatus != TRAN_OK) {
                display_status(data->s_transtatus);
                return -1;
  }

  char buf[32];
  position(threadP,  52, 6); format_int(buf, 3, data->s_D_ID);   threadP->write(buf, 2);
  position(threadP,  33,11); format_int(buf, 6, data->s_C_W_ID); threadP->write(buf, 4);
  position(threadP,  54,11); format_int(buf, 3, data->s_C_D_ID); threadP->write(buf, 2);
  position(threadP,   7, 4); threadP->write( data->s_H_DATE );
  position(threadP,   1, 7); threadP->write( data->s_W_STREET_1);
  position(threadP,  42, 7); threadP->write( data->s_D_STREET_1);
  position(threadP,   1, 8); threadP->write( data->s_W_STREET_2);
  position(threadP,  42, 8); threadP->write( data->s_D_STREET_2);
  position(threadP,   1, 9); threadP->write( data->s_W_CITY);
  position(threadP,  22, 9); threadP->write( data->s_W_STATE);
  position(threadP,  25, 9); format_zip(buf, 10, data->s_W_ZIP); threadP->write(buf, 10);
  position(threadP,  42, 9); threadP->write( data->s_D_CITY);
  position(threadP,  63, 9); threadP->write( data->s_D_STATE);
  position(threadP,  66, 9); format_zip(buf, 10, data->s_D_ZIP); threadP->write(buf, 10);
  position(threadP,  11,11); format_int( buf, 6, data->s_C_ID); threadP->write(buf, 4);
  position(threadP,   9,12); threadP->write( data->s_C_FIRST);
  position(threadP,  26,12); threadP->write( data->s_C_MIDDLE);
  position(threadP,  29,12); threadP->write( data->s_C_LAST);
  position(threadP,  58,12); format_date(buf, 10, data->s_C_SINCE); threadP->write( buf, 10);
  position(threadP,   9,13); threadP->write( data->s_C_STREET_1);
  position(threadP,  58,13); threadP->write( data->s_C_CREDIT);
  position(threadP,   9,14); threadP->write( data->s_C_STREET_2);
  position(threadP,  58,14); format_float(buf, 6, 2, data->s_C_DISCOUNT/100); threadP->write(buf, 6);
  position(threadP,   9,15); threadP->write( data->s_C_CITY);
  position(threadP,  30,15); threadP->write( data->s_C_STATE);
  position(threadP,  33,15); format_zip(buf, 10, data->s_C_ZIP); threadP->write(buf, 10);
  position(threadP,  58,15); format_phone(buf, 18, data->s_C_PHONE ); threadP->write(buf, 18);
  position(threadP,  17,17); format_money( buf, 15, data->s_H_AMOUNT); threadP->write(buf, 14);
  position(threadP,  55,17); format_money( buf, 16, data->s_C_BALANCE); threadP->write(buf, 15);
  position(threadP,  17,18); format_money( buf, 15, data->s_C_CREDIT_LIM); threadP->write(buf, 14);

  if (data->s_C_CREDIT[0] == 'B' && data->s_C_CREDIT[1] == 'C') {
                int i, size = strlen((char *)data->s_C_DATA);
                for (i = 0; i < 4; i++) {
                  position(threadP, 12,20+i);
                  threadP->write(data->s_C_DATA, (size > 50)?50:size);
                  size -= 50;
                  if (size <= 0) break;

                }
  }

  return 0;
}

/*****************************************************************************
OrderStatus
*****************************************************************************/
OrderStatus::OrderStatus(User_data *udP, Thread_data *threadP) : Screen(udP, threadP) {
  tran_type = ORDERSTATUS_SERVICE;
  dataptr = data = new OrderStatus_data;
  data_len = sizeof(OrderStatus_data);

  status_x=1;
  status_y=25;

  int pos = 0;
  screen          = static_screen;
  empty_fields    = static_empty_fields;
#ifdef LOCAL_SESSION_DATA
  session_data = new char[static_session_data_len+1];
  sprintf(session_data, "%s%5d", POS(12,4), user_dataP->warehouse);
#else
  session_data    = static_session_data;
  sprintf(session_data, "%s%5d", POS(12,4), user_dataP->warehouse);
```

```
#endif
   screen_len      = static_screen_len;
   empty_fields_len= static_empty_fields_len;
   session_data_len= static_session_data_len;

   fields = new Field*[4];
   fields[pos++]= genfield(threadP, 29, 4, 2, &data->s_D_ID ); /* District */
   fields[pos++]= genfield(threadP, 11, 5, 4, &data->s_C_ID ); /* Customer ID */
   fields[pos++]= genfield(threadP, 44, 5, 16, (char *)data->s_C_LAST ); /* Customer Name */
   fields[pos++]= NULL;
#if defined(USE_SMART_FIELDS)
   fields[1]->ok_func= (int(*)(void*))pos_zero;
   fields[1]->ok_data= &fields[2]->pos;
   fields[2]->ok_func= (int(*)(void*))pos_zero;
   fields[2]->ok_data= &fields[1]->pos;
#endif
};

int OrderStatus::validate() {
   if (!fields[0]->pos) {
                pos=0;
                message(threadP, "District ID is a required field");
                return 0;
   }
   if (fields[1]->pos) {
#if defined(USE_BYNAME)
                data->s_byname= 0;
#endif
   } else if (fields[2]->pos) {
#if defined(USE_BYNAME)
                data->s_byname= 1;
#endif
   } else {
                pos=1;
                message(threadP, "Customer ID or Name is required");
                return 0;
   }

   data->s_W_ID = user_dataP->warehouse;

   return 1;
}

int OrderStatus::respond() {
   display_status(data->s_transtatus);
   if (data->s_transtatus != TRAN_OK)
                return -1;

   char buf[32];

   position(threadP, 11, 5); format_int(buf, 6, data->s_C_ID); threadP->write(buf, 4);
   position(threadP, 24, 5); threadP->write(data->s_C_FIRST);
   position(threadP, 41, 5); threadP->write(data->s_C_MIDDLE);
   position(threadP, 44, 5); threadP->write(data->s_C_LAST);
   position(threadP, 15, 6); format_money(buf, 11, data->s_C_BALANCE); threadP->write(buf, 10);
   position(threadP, 15, 8); format_int(buf, 10, data->s_O_ID); threadP->write(buf, 9);
   position(threadP, 38, 8); format_date(buf, 19, data->s_O_ENTRY_D); threadP->write(buf);
   if (data->s_O_CARRIER_ID > 0) {
                position(threadP, 76, 8);
                format_int(buf, 3, data->s_O_CARRIER_ID);
                threadP->write(buf, 2);
   }

   for (int i=0; i < data->s_ol_cnt; i++) {
                position(threadP,  3, i+10);
                format_int(buf, 6, data->item[i].s_OL_SUPPLY_W_ID);
                threadP->write(buf, 5);

                position(threadP, 14, i+10);
                format_int(buf, 7, data->item[i].s_OL_I_ID);
                threadP->write(buf, 6);

                position(threadP, 25, i+10);
                format_int(buf, 3, data->item[i].s_OL_QUANTITY);
                threadP->write(buf, 2);

                position (threadP, 32, i+10);
                format_money(buf, 10, data->item[i].s_OL_AMOUNT);
                threadP->write(buf, 9);

                position (threadP, 47, i+10);
                format_date(buf, 20, data->item[i].s_OL_DELIVERY_D);
                threadP->write(buf, 19);
   }

   return 0;
}

/*****************************************************************************
Delivery
*****************************************************************************/
Delivery::Delivery(User_data*udP, Thread_data *threadP) : Screen(udP, threadP) {
   tran_type = DELIVERY_SERVICE;
   dataptr = data = new Delivery_data;
   data_len = sizeof(Delivery_data);

   status_x = 1;
   status_y = 8;

   int pos = 0;
   screen          = static_screen;
   empty_fields    = static_empty_fields;
#ifdef LOCAL_SESSION_DATA
   session_data = new char[static_session_data_len+1];
   sprintf(session_data, "%s%5d", POS(12,4), user_dataP->warehouse);
#else
   session_data    = static_session_data;
   sprintf(session_data, "%s%5d", POS(12,4), user_dataP->warehouse);
#endif
   screen_len      = static_screen_len;
   empty_fields_len= static_empty_fields_len;
   session_data_len= static_session_data_len;

   fields = new Field*[2];
   fields[pos++]= genfield(threadP, 17, 6, 2, &data->s_O_CARRIER_ID ); /* Carrier Number */

   fields[pos++]= NULL;
};

int Delivery::validate(){
   if (!fields[0]->pos) {
                pos=0;
                message(threadP, "Carrier ID is a required field");
                return 0;
   }
   time((time_t *)&(data->s_queued_time));

   data->s_W_ID = user_dataP->warehouse;

   return 1;
}

int Delivery::respond() {
   if (data->s_transtatus == TRAN_OK) {
                position(threadP, status_x, status_y);
                threadP->write("Execution Status: Delivery has been queued");
   } else {
                display_status(data->s_transtatus);
                return -1;
   }
   return 0;
}

/*****************************************************************************
StockLevel
*****************************************************************************/
StockLevel::StockLevel(User_data *udP, Thread_data *threadP) : Screen(udP, threadP) {
   tran_type = STOCKLEVEL_SERVICE;
   dataptr = data = new StockLevel_data;
   data_len = sizeof(StockLevel_data);

   status_x = 1;
   status_y = 10;

   int pos = 0;
   screen          = static_screen;
   empty_fields    = static_empty_fields;
   session_data    = static_session_data;
#ifdef LOCAL_SESSION_DATA
   session_data = new char[static_session_data_len+1];
   sprintf(session_data, "%s%5d%s%2d", POS(12,4), user_dataP->warehouse,
                                                                        POS(29,4),
user_dataP->district);
#else
   session_data    = static_session_data;
   sprintf(session_data, "%s%5d%s%2d", POS(12,4), user_dataP->warehouse,
                                                                        POS(29,4),
user_dataP->district);
#endif
   screen_len      = static_screen_len;
   empty_fields_len= static_empty_fields_len;
   session_data_len= static_session_data_len;

   fields = new Field*[2];
   fields[pos++]= genfield(threadP, 24, 6, 2, &data->s_threshold ); /* Threshold */
   fields[pos++]= NULL;
};

int StockLevel::validate() {
   if (data->s_threshold <= 0) {
                pos=0;
                message(threadP, "A positive non-zero threshold is required");
                return 0;
   }
   data->s_W_ID = user_dataP->warehouse;
   data->s_D_ID = user_dataP->district;

   return 1;
}

int StockLevel::respond() {
   display_status(data->s_transtatus);
   if (data->s_transtatus != TRAN_OK)
                return -1;

   position(threadP, 12, 8);
   char buf[5];
   format_int(buf, 4, data->s_low_stock);
```

```
    threadP->write(buf, 4);

    return 0;
}

/*****************************************************************************
 perform
*****************************************************************************/
int NewOrder::process() {
    if (tran_type == NULL)
                return 0;

    if (encina.tran(data, threadP->contextP, tran_type) < 0) {
                return -1;
    }
    return 0;
}

int Payment::process() {
    if (tran_type == NULL)
                return 0;

    if (encina.tran(data, threadP->contextP, tran_type) < 0) {
                return -1;
    }
    return 0;
}

int StockLevel::process() {
    if (tran_type == NULL)
                return 0;

    if (encina.tran(data, threadP->contextP, tran_type) < 0) {
                return -1;
    }
    return 0;
}

int OrderStatus::process() {
    if (tran_type == NULL)
                return 0;

    if (encina.tran(data, threadP->contextP, tran_type) < 0) {
                return -1;
    }
    return 0;
}

int Delivery::process() {
    if (tran_type == NULL)
       return 0;

    if (encina.tran(data, threadP->contextP, tran_type) < 0) {
       return -1;
    }
    return 0;
}


int Screen::process() {
    if (tran_type == NULL)
                return 0;
    return 0;
}


/*****************************************************************************
Login
*****************************************************************************/
Login::Login(User_data *udP, Thread_data *threadP) : Screen(udP, threadP) {
    tran_type = NULL;
    status_x=1;
    status_y=24;

    dataptr = NULL;
    data_len = 0;

    int pos = 0;
    screen          = static_screen;
    screen_len      = static_screen_len;
    empty_fields    = static_empty_fields;
    empty_fields_len= static_empty_fields_len;

    fields = new Field*[3];
    fields[pos++] = genfield(threadP, 16,  5,  5, &(udP->warehouse) ); //Warehouse
    fields[pos++] = genfield(threadP, 34,  5,  2, &(udP->district)  ); //District
    fields[pos++] = NULL;
};
int Login::validate() {
    if (!fields[0]->pos) {
                pos=0;
                message(threadP, "Warehouse ID is a required field");
                return 0;
    }
    if (!fields[1]->pos) {
                pos=1;
                message(threadP, "District ID is a required field");
                return 0;
    }
    return 1;
}


/*****************************************************************************
Menu
*****************************************************************************/
Menu::Menu(User_data *udP, Thread_data *threadP) : Screen(udP, threadP) {
    tran_type = NULL;
    status_x=1;
    status_y=24;

    int pos = 0;
    screen          = static_screen;
    screen_len      = static_screen_len;
    empty_fields    = NULL;
    empty_fields_len= 0;


    fields = NULL;
};

/*****************************************************************************
Static data
*****************************************************************************/
char const * const blanks      = "                                        ";
char const * const underscores = "________________________________________";
char const * const backspaces  = "\b\b\b\b\b\b\b\b\b\b";

/*****************************************************************************
Utility Functions
*****************************************************************************/
static int string_empty(char const *data) {
    return data[0] == 0;
}
static int pos_zero(int const *val) {
    return *val == 0;
}
static int pos_nonzeros(int const **val) {
    int const **ptr;
    for (ptr = val; *ptr; ptr++) {
                if (**ptr == 0)
                    return 0;
    }
    return 1;
}

int position(int x, int y, char *buf) {
    int pos = 0;
    buf[pos++] = ESCc;
    buf[pos++] = '[';
    if (y >= 10) buf[pos++] = (y / 10) + '0';
    buf[pos++] = (y % 10) + '0';
    buf[pos++] = ';';
    if (x >= 10) buf[pos++] = (x / 10) + '0';
    buf[pos++] = (x % 10) + '0';
    buf[pos++] = 'H';
    buf[pos++] = 0;
    return 0;
}

int position(InOut *threadP, int x, int y) {
    char buf[16];
    position(x, y, buf);
    threadP->write(buf);
    return 0;
}

static int clear_eos(InOut *threadP) {
    threadP->write (ESC "[J");
    return 0;
}


int message(InOut *threadP, char const *text, int need_flush) {
    position(threadP, 1,25);
    threadP->write(text);
    clear_eos(threadP);
    if (need_flush)
      threadP->flush();
    return 0;
}

static int clear_eos(char *buf) {
    buf[0] = ESCc;
    buf[1] = '[';
    buf[2] = 'J';
    return 0;
}
```

screen.h

```
/* (C)1997 IBM Corporation */
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <ctype.h>
```

```
#ifndef WIN32
#include <termios.h>
#endif
#include <time.h>

#include "field.h"
#include "inout.h"
#include "tpcc.h"

extern int position(int x, int y, char *buf);
extern int position(InOut *ioP, int x, int y);
extern int message(InOut *ioP, char const *text, int need_flush=1);


class User_data {
public:
   int warehouse;
   int district;
};

class Thread_data : public InOut {
 public:
   void           *contextP;
   Thread_data(int infd, int outfd, void *conP) : InOut(infd, outfd), contextP(conP) {};
};

class Screen {
protected:
   static char const end_str[];
   static int       end_str_len;
          int       has_data;
   void       *dataptr;
   char       *tran_type;
   char const *screen;
   char const *empty_fields;
   char       *session_data;
   int        screen_len;
   int        session_data_len;
   int        empty_fields_len;
   int        pos;
   int        status_x, status_y;
   int        data_len;
   Thread_data                  *threadP;

public:
   User_data  *user_dataP;
   Field **fields;
   virtual char const *isa() { return "Screen"; };
   virtual int reset();
   virtual int present();
   virtual int present_empty_fields();
   virtual int process();
   virtual int user_input();
   virtual int validate()           { return 1; };
   virtual int respond()            { return 0; };
   int handle();
   int display_status(int status);
   Screen(User_data *udP, Thread_data *thrP) {
                  user_dataP = udP;
                  threadP = thrP;
                  has_data = 0;
                  pos = 0;
                  fields = NULL;
                  screen = empty_fields= session_data = NULL;
                  screen_len = session_data_len = empty_fields_len= 0;
   };
   virtual ~Screen();
};

class Login : public Screen {
protected:
   static char const static_screen[];
   static char const static_empty_fields[];
   static char static_session_data[];
   static int  static_screen_len;
   static int  static_empty_fields_len;
   static int  static_session_data_len;
public:
   int validate();
   Login::Login(User_data *udP, Thread_data *thrP);
};

class NewOrder : public Screen {
protected:
   static char const static_screen[];
   static char const static_empty_fields[];
   static char static_session_data[];
   static int  static_screen_len;
   static int  static_empty_fields_len;
   static int  static_session_data_len;
   int start_field;
   void swap_fields(int i, int j);
public:
   NewOrder_data *data;

   int reset();
   NewOrder::NewOrder(User_data *udP, Thread_data *thrP);
   int validate();
   int process();
   int respond();
};

class Payment : public Screen {
protected:
   static char const static_screen[];
   static char const static_empty_fields[];
   static char static_session_data[];
   static int  static_screen_len;
   static int  static_empty_fields_len;
   static int  static_session_data_len;
public:
   Payment_data *data;
   int validate();
   int process();
   int respond();
   Payment(User_data *udP, Thread_data *thrP);
};

class OrderStatus : public Screen {
protected:
   static char const static_screen[];
   static char const static_empty_fields[];
   static char static_session_data[];
   static int  static_screen_len;
   static int  static_empty_fields_len;
   static int  static_session_data_len;
public:
   OrderStatus_data *data;
   int validate();
   int process();
   int respond();

   OrderStatus(User_data *udP, Thread_data *thrP);
};

class Delivery : public Screen {
protected:
   static char const static_screen[];
   static char const static_empty_fields[];
   static char static_session_data[];
   static int  static_screen_len;
   static int  static_empty_fields_len;
   static int  static_session_data_len;
public:
   Delivery_data *data;
   int validate();
   int process();
   int respond();

   Delivery(User_data *udP, Thread_data *thrP);
};

class StockLevel : public Screen {
protected:
   static char const static_screen[];
   static char const static_empty_fields[];
   static char static_session_data[];
   static int  static_screen_len;
   static int  static_empty_fields_len;
   static int  static_session_data_len;
public:
   StockLevel_data *data;
   int validate();
   int process();
   int respond();

   StockLevel(User_data *udP, Thread_data *thrP);
};

class Menu : public Screen {
protected:
   static char const static_screen[];
   static char const static_empty_fields[];
   static char static_session_data[];
   static int  static_screen_len;
   static int  static_empty_fields_len;
   static int  static_session_data_len;
public:
   Menu(User_data *udP, Thread_data *thrP);
};
```

screen_data.C

```
/* (C)1997 IBM Corporation */
#include "screen.h"


char const NewOrder::static_screen[] =
   POS( 1, 3) CLEAR_EOS
   POS(36, 3) "New Order"
   POS( 1, 4) "Warehouse"
   POS(19, 4) "District:"
   POS(55, 4) "Date:"
   POS( 1, 5) "Customer:"

   POS(19, 5) "Name:"
```

```
    POS(44, 5) "Credit:"
    POS(57, 5) "Disc.:"
    POS( 1, 6) "Order Number:"
    POS(25, 6) "Number of Lines:"
    POS(52, 6) "W_Tax:"
    POS(67, 6) "D_Tax:"
    POS( 2, 8) "Supp_W Item_Num Item_Name"
    POS(44, 8) "Qty  Stock  B/G   Price   Amount"
;

char const NewOrder::static_empty_fields[]=
    POS(29, 4) "____" /* District */
    POS(12, 5) "____" /* Customer */

    POS( 3, 9) "_____"
    POS(10, 9) "_______"
    POS(45, 9) "__"
    POS( 3,10) "_____"
    POS(10,10) "_______"
    POS(45,10) "__"
    POS( 3,11) "_____"
    POS(10,11) "_______"
    POS(45,11) "__"
    POS( 3,12) "_____"
    POS(10,12) "_______"
    POS(45,12) "__"
    POS( 3,13) "_____"
    POS(10,13) "_______"
    POS(45,13) "__"
    POS( 3,14) "_____"
    POS(10,14) "_______"
    POS(45,14) "__"
    POS( 3,15) "_____"
    POS(10,15) "_______"
    POS(45,15) "__"
    POS( 3,16) "_____"
    POS(10,16) "_______"
    POS(45,16) "__"
    POS( 3,17) "_____"
    POS(10,17) "_______"
    POS(45,17) "__"
    POS( 3,18) "_____"
    POS(10,18) "_______"
    POS(45,18) "__"
    POS( 3,19) "_____"
    POS(10,19) "_______"
    POS(45,19) "__"
    POS( 3,20) "_____"
    POS(10,20) "_______"
    POS(45,20) "__"
    POS( 3,21) "_____"
    POS(10,21) "_______"
    POS(45,21) "__"
    POS( 3,22) "_____"
    POS(10,22) "_______"
    POS(45,22) "__"
    POS( 3,23) "_____"
    POS(10,23) "_______"
    POS(45,23) "__"
;

char NewOrder::static_session_data[] =
    POS(12,4) "#####" /* Warehouse Id */
;

int NewOrder::static_screen_len = sizeof(NewOrder::static_screen) - 1;
int NewOrder::static_empty_fields_len= sizeof(NewOrder::static_empty_fields)- 1;
int NewOrder::static_session_data_len = sizeof(NewOrder::static_session_data) - 1;

/* Payment */
char const Payment::static_screen[] =
    POS( 1, 3) CLEAR_EOS
    POS( 38,3) "Payment"
    POS( 1,4) "Date:"
    POS( 1,6) "Warehouse:"
    POS( 42,6) "District:"
    POS( 1,11) "Customer:"
    POS( 17,11) "Cust-Warehouse:"
    POS( 39,11) "Cust-District:"
    POS( 1,12) "Name:"
    POS( 50,12) "Since:"
    POS( 50,13) "Credit:"
    POS( 50,14) "%Disc:"
    POS( 50,15) "Phone:"
    POS( 1,17) "Amount Paid:"
    POS( 37,17) "New Cust-Balance:"
    POS( 1,18) "Credit Limit:"
    POS( 1,20) "Cust-Data:"
;
char const Payment::static_empty_fields[]=
    POS( 52, 6) "__" /* District */
    POS( 11,11) "____" /* Customer # */
    POS( 33,11) "_____" /* Cust-Warehouse */
    POS( 54,11) "__" /* Cust-District */
    POS( 29,12) "________________" /* Name */
    POS( 23,17) "________" /* Amount Paid */
;

char Payment::static_session_data[] =
    POS( 12,6) "#####" /* Warehouse */
;

int Payment::static_screen_len= sizeof(Payment::static_screen)- 1;
int Payment::static_empty_fields_len= sizeof(Payment::static_empty_fields)- 1;
int Payment::static_session_data_len= sizeof(Payment::static_session_data)- 1;

/* Order Status */
char const OrderStatus::static_screen[] =
    POS( 1, 3) CLEAR_EOS
    POS(35, 3) "Order-Status"
    POS( 1, 4) "Warehouse:"
    POS(19, 4) "District:"
    POS( 1, 5) "Customer:"
    POS(18, 5) "Name:"
    POS( 1, 6) "Cust-Balance:"
    POS( 1, 8) "Order-Number"
    POS(26, 8) "Entry-Date:"
    POS(60, 8) "Carrier-Number:"
    POS( 1, 9) "Supply-W"
    POS(14, 9) "Item-Num"
    POS(25, 9) "Qty"
    POS(33, 9)"Amount"
    POS(45, 9) "Delivery-Date"
;
char const OrderStatus::static_empty_fields[] =
    POS(29, 4) "__" /* District */
    POS(11, 5) "____" /* Customer ID */
    POS(44, 5) "________________" /* Customer Name */
;

char OrderStatus::static_session_data[] =
    POS(12, 4) "#####" /* Warehouse */
;
int OrderStatus::static_screen_len = sizeof(OrderStatus::static_screen) - 1;
int OrderStatus::static_empty_fields_len= sizeof(OrderStatus::static_empty_fields)- 1;
int OrderStatus::static_session_data_len = sizeof(OrderStatus::static_session_data) - 1;

/* Delivery */
char const Delivery::static_screen[] =
    POS(1,3) CLEAR_EOS
    POS( 38,3) "Delivery"
    POS( 1,4) "Warehouse:"
    POS( 1,6) "Carrier Number:"
;
char const Delivery::static_empty_fields[]=
    POS( 17,6) "__" /* Carrier Number */
;
char Delivery::static_session_data[]=
    POS( 12, 4) "#####" /* Warehouse */
;
int Delivery::static_screen_len= sizeof(Delivery::static_screen)- 1;
int Delivery::static_empty_fields_len= sizeof(Delivery::static_empty_fields)- 1;
int Delivery::static_session_data_len= sizeof(Delivery::static_session_data)- 1;

/* Stock level */
char const StockLevel::static_screen[] =
    POS( 1, 3) CLEAR_EOS
    POS(35, 3) "Stock-Level"
    POS( 1, 4) "Warehouse:"
    POS(19, 4) "District:"
    POS( 1, 6) "Stock Level Threshold:"
    POS( 1, 8) "Low Stock:"
;

char const StockLevel::static_empty_fields[]=
    POS( 24,6) "__" /* Threshold */
;

char StockLevel::static_session_data[] =
    POS( 12,4) "#####" /* Warehouse */
    POS( 29,4) "##"   /* District */
;
int StockLevel::static_screen_len= sizeof(StockLevel::static_screen) - 1;
int StockLevel::static_empty_fields_len= sizeof(StockLevel::static_empty_fields)- 1;
int StockLevel::static_session_data_len= sizeof(StockLevel::static_session_data)- 1;

/* Login */
char const Login::static_screen[] =
    POS( 1, 1) CLEAR_EOS
    POS(30, 3) "Please login."
    POS( 5, 5) "Warehouse:"
    POS(24, 5) "District:"
;
char const Login::static_empty_fields[]=
    POS( 16,5) "_____" /* Warehouse */
    POS( 34,5) "__" /* District */
;
int Login::static_screen_len= sizeof(Login::static_screen) - 1;
int Login::static_empty_fields_len= sizeof(Login::static_empty_fields)- 1;

/* Menu */
char const Menu::static_screen[] =
    POS(1, 1) CLEAR_EOS
    "(1)New-Order (2)Payment (3)Order-Status (4)Delivery (5)StockLevel (9)Exit"
;

int Menu::static_screen_len= sizeof(Menu::static_screen) - 1;
```

```
/* end string */
char const Screen::end_str[] = "\033[H\n";
int Screen::end_str_len = sizeof(Screen::end_str) - 1;
```

server.c

```
/*
* server.c
*
* $Revision: 1.11 $
* $Date: 1999/05/06 21:28:30 $
* $Log:          server.c,v $
*
* $TALog: server.c,v $
* Revision 1.11  1999/05/06  21:28:30  oz
* - Removed all the .. from the includes
* - Added -I.. to the makefiles instead
* - Moved all the thread related code and connection
*   selection to serverMon.c
* [from r1.10 by delta oz-24309-TPCC-add-oracle8.1-code, r1.5]
*
* Revision 1.10  1999/04/19  20:14:48  oz
* - Moved all the simulated code to server.c
* - Created nulldb.c for compilation with no DB
* [from r1.8 by delta oz-24331-TPCC-move-sim-code-to-common-file, r1.1]
*
* Revision 1.9  1999/04/14  18:11:56  wenjian
* Make changes so that the web client data structures for transactions
* are same as the data structures used in SQL server. It is an important
* change to integrate with MS TPCC kit. It will also avoid copyin/copyout
* for each transaction.
* [from r1.8 by delta wenjian-24134-TPCC-make-client-data-structure-same-as-server, r1.1]
*
* Revision 1.8  1998/12/11  16:37:58  wenjian
* Move some common functions from client/client_utils.c to common/tpcc_utils.c.
* In this version, we only move time_diff_ms(). Need some work in order to
* move other functions like ERROUT.
*
* - Move time_diff_ms() to common/tpcc_utils.c
* [from r1.7 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.2]
*
* Revision 1.7  1998/12/11  16:14:20  wenjian
* Add code for checking statistic data in a single variable and collecting
* statistic data based on iStatsFrequency.
*
* - Add time_diff_ms()
* [from r1.6 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.1]
*
* Revision 1.6  1998/11/09  16:59:47  wenjian
* In this revision, most of the changes are related to the directory of header
* files after directory reorganization. Other changes include adding or removing
* files to put them in the right directories. Makefiles are written for NT
* platform so that nmake is working on NT now. Need a top level Makefile for all
* the directories.
* [from r1.5 by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.2]
*
* Revision 1.5  1998/11/09  14:48:24  wenjian
* In an effort to make a new directory structure for TPCC, this delta
* creates two directories: tpcc/client and tpcc/server. All the files
* for this revision are copied from tpcc/sp-tpcc without any change.
* Further change may be needed for some files due to the change of
* the directory structure.
* [added by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.1]
*
* Revision 1.13  1998/11/06  16:10:56  wenjian
* - Change num_mults from 5 to 20
* [from r1.12 by delta wenjian-23646-TPCC-clean-up-source-code, r1.1]
*
* Revision 1.12  1998/10/22  16:25:12  wenjian
* Multi-threaded version.
*
* - Define deliLog for the output of delivery server
* - Stop printing to stderr in err_printf() since it seems to be too
*   expensive on NT. Print to file instead.
* [from r1.11 by delta wenjian-23529-TPCC-integrate-with-SQL-server, r1.2]
*
* Revision 1.11  1998/06/17  15:05:48  wenjian
* Somehow, read and write didn't work for socket on NT, although they
* are supposed to work. As a work-around way, use recv and send for
* NT in this revision. We may change them back if the problem is gone.
*
* Use recv and send for socket read and write.
* [from r1.10 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.2]
*
* Revision 1.10  1998/02/17  22:07:06  wenjian
* Define macros to deal with the different function names on NT
* [from r1.9 by delta wenjian-21750-TPCC-changes-for-porting-on-NT, r1.1]
*
* Revision 1.9  1998/01/23  15:08:48  oz
* - Updated the SP TPCC directory to the latest files used
*   during the SP tpcc audit.
* [from r1.8 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
*
*
*/
/*
*
*     TPCC Server
*
*  There are currently three versions of the TPCC benchmark
*  implemented here: An Encina monitor based benchmark,
*  an Encina Toolkit based benchmark and a DCE only benchmark.
*
*  This file, server.c, contains all the code that is common to
*  all the versions.  Each server has its own main file:
*  serverMon.c for the monitor server, serverTK.c for the toolkit
*  server and serverDce.c for the dce server.
*
*  Each server is comprised of three main modules: the serverspecifc
*  one (mentioned above), the common one, in this file, and the
*  server part, which is in the SQL files DBInfo.ec, dbInit.ec,
*  delivery.ec, newOrder.ec, orderStatus.ec, payment.ec, stockLevel.ec.
*/
#include <sys/types.h>
#ifndef WIN32
#include <sys/socket.h>
#include <sys/errno.h>
#else
#include <winsock.h>
#include <io.h>
#endif
#include <fcntl.h>
#include <stdio.h>
#include <stdarg.h>
#include <stdlib.h>
#include <string.h>
#if defined (solaris)
#include <dce/pthread.h>
#else  /* solaris */
#include <pthread.h>
#endif
#include <utils/trace.h>
#include <tpm/mon.h>

#include "common/utilities.h"
#include "server.h"
#include "common/tpcc_type.h"
#include "common/do_tpcc.h"
#include <time.h>

#define DEFINE_SERVER_DEBUG
#include "serverDebug.h"

#if defined(solaris)
extern int errno;
#endif

#ifdef WIN32
#define O_RDONLY _O_RDONLY
#define  read(A,B,C)    recv(A,B,C,0)
#define open     _open
#define close    _close
#endif

#define SIM_ERROR_CODE TPCC_SUCCESS

#ifdef WIN32
#define RANDOM rand
#else
#define RANDOM random
#endif

#define TPCC_HOME    "/tmp"
#define TIME_PREFIX_LEN 50
/* extern char sys_errlist[]; */
extern time_diff_ms(struct timeval *, struct timeval *);

void dprint(char *format, ...);
/*
*  Global variables common to all types of servers
*/
FILE *server_logtrans = NULL;
FILE *deliLog = NULL;
int logtrans = -1;
FILE *dvry_log = NULL;      /*  FILE structure for delivery log */
int dvry_log_fd = -1;       /*  File descriptor for delivery log*/
int status_log = -1;        /*  File descriptor for status log   */
FILE *deliveryLog = NULL;
FILE *deliveryOut = NULL;
int serverIdNumber = 0;     /*  The ID of the server
                                *  This is used to identify output
                                */
int serverPid = 0;
int num_mults = 80;         /*  The number of times the matrices are
                                *  multiplied (in order to spend some time)
                                */

int server_null_test = 0;
int server_init = 0;        /*  The time (in seconds) the test started
                                *  This is used by the deferred delivery
                                *  which reports its times as elapsed time
                                *  since start time
                                */
int null_with_sleep = 1;  /*  Sleep for some time when simulatingtrans */
```

```
void err_printf(char *format, ...);
void logprintf(char *format, ...);

void open_log_files()
{
  /* open DVRY_LOG to keep delivery transactions logs*/
  char logname[MAX_STR_LEN], fname[MAX_STR_LEN];
  char buffer[MAX_STR_LEN];
  char *tpcc_home;
  char *log_dir;
  int  bytes;
  int  current_fp;
  int  current;

  log_dir = getenv("DELIVERY_LOGS");
  if (log_dir == NULL) {
                fprintf(stderr, "DELIVERY_LOGS not specified, using %s\n",
                                TPCC_HOME);
                log_dir = TPCC_HOME;
  }

/*
  sprintf(buffer, "%s/status.%d", log_dir, getpid());
  status_log = creat(buffer, 0666 );
*/

  tpcc_home = getenv("TPCC_HOME");
  if (tpcc_home == NULL) {
                fprintf(stderr, "TPCC_HOME not specified, using /tmp\n");
                tpcc_home = "/tmp";
  }

  sprintf(fname, "%s/CURRENT", tpcc_home );
  current_fp = open(fname, O_RDONLY);
  bytes = read(current_fp, buffer, MAX_STR_LEN);

  if (bytes == -1) {
                fprintf(stderr, "Could not read CURRENT file.\n");
                exit(1);
  }
  buffer[bytes] = '\0';
  current = atoi(buffer);
  close(current_fp);

  dvry_log = NULL;
}


/*
* logprintf() -- variable argument function used to print error
*                                and debug statements. Function is called when
*                                any of the debug macros (defined inserverDebug.h)
*                                are used.
*/


/*
*  get_thread_id
*      A function that returns the thread ID of the current thread
*/
int get_thread_id()
{
  pthread_t thread = pthread_self();
  int thread_id = pthread_getunique_np(&thread);
  return(thread_id);
}

void print_time_prefix(FILE*file)
{
  time_t cur_timet;
  char time_str[30];

  cur_timet = time(&cur_timet);
  strftime(time_str, 29, "%X", localtime(&cur_timet));

  fprintf(file,"%4d %5d %4d %s - ",
                  serverIdNumber, serverPid, get_thread_id(), time_str);
}


char *get_time_prefix(char*buffer)
{
  time_t cur_timet;
  char time_str[30];
  int len;

  cur_timet = time(&cur_timet);
  strftime(time_str, 29, "%X", localtime(&cur_timet));

  len = sprintf(buffer, "%4d %5d %4d %s - ",
                                  serverIdNumber, serverPid, get_thread_id(), time_str);
  if (len >= TIME_PREFIX_LEN) {
    fprintf(stderr, "TIME_PREFIX_LEN (%d) too small: %d\n",
                                TIME_PREFIX_LEN, len);
    exit(12);
  }
  return(buffer);
}

void logprintf(char *format, ...)
{
  char formatBuffer[200];
  char *fmt = formatBuffer;
  int fmtLen;
  va_list ap;
  va_start(ap, format);

  fmtLen = TIME_PREFIX_LEN + strlen(format) + 2;
  if (fmtLen > sizeof(formatBuffer)) {
     fmt = (char *)malloc(fmtLen);
  }
  get_time_prefix(fmt);
  strcat(fmt, format);
  if (server_logtrans) {
                vfprintf(server_logtrans, fmt, ap);
                fflush(server_logtrans);
  } else {
                vfprintf(stderr, fmt, ap);
  }
  if (fmt != formatBuffer) free(fmt);
  va_end(ap);
}

void err_printf(char *format, ...)
{
  char formatBuffer[200];
  char *fmt = formatBuffer;
  int fmtLen;
  char timeBuffer[128];
  va_list ap;
  va_start(ap, format);

  fmtLen = TIME_PREFIX_LEN + strlen(format) + 2;
  if (fmtLen > sizeof(formatBuffer)) {
     fmt = (char *)malloc(fmtLen);
  }
  get_time_prefix(fmt);
  strcat(fmt, format);
  if (server_logtrans) {
     vfprintf(server_logtrans, fmt, ap);
     fflush(server_logtrans);
  } else {
                vfprintf(stderr, fmt, ap);
  }

  if (fmt != formatBuffer) free(fmt);
  va_end(ap);
}


/*
* dprint() -- variable argument function used to print debug
*                        statements; for use with DPRINT macro.
*/

void dprint(char *format, ...)
{
  va_list ap;
  va_start(ap, format);

  print_time_prefix(stderr);
  vfprintf(stderr, format, ap);

  va_end(ap);
}

/***  Code that has to do with null servers and simulated DBs  ***/

void mat_mult(int);

#define ROWS 5
#define COLS 5

double matrix_a[ROWS][COLS] = {
 {1.2, 3.4, 2.3, 4.6, 5.2},
 {2.3, 4.5, 1.2, 9.4, 3.1},
 {3.4, 5.2, 3.8, 6.5, 1.6},
 {1.2, 5.3, 6.1, 2.9, 3.8},
 {2.4, 1.2, 3.4, 7.2, 1.0}
};
double matrix_b[ROWS][COLS] = {
 {3.4, 5.9, 2.8, 3.4, 5.6},
 {7.2, 9.3, 4.6, 5.2, 1.3},
 {6.4, 5.2, 8.3, 9.4, 2.3},
 {7.2, 3.4, 6.9, 8.1, 2.3},
 {2.3, 4.5, 7.2, 3.4, 5.8}
};

/*  Num of ms to add to RT  */
static int rt_increment = 0;

/* Num of ms to add to rt_increment after a certain time. */
static int more_srv_work = 0;

/* how often (in second) to add more_srv_work to rt_increment*/
static int period_to_add_rt = 7*60;
```

```
/* how often (in second) to check if there is transaction */
static int period_to_check_tran = 10;

static struct timespec *get_wait_time(struct timespec *timeP, int tran)
{
  int ran = RANDOM() % 1000;
  int wait;

  if (0) {
     if (ran > 998) {
                    timeP->tv_sec = 10;
     } else if (ran > 990) {
                    timeP->tv_sec = 5;
     } else if (ran > 970) {
        timeP->tv_sec = 1;
     } else {
                    timeP->tv_sec = 0;
     }
     timeP->tv_nsec = 50000000;
     if (tran == NEWO_TRANS) {
                    timeP->tv_nsec *= 2;
                    timeP->tv_sec *= 2;
     }
  } else {
                int time_ms = 0;
                if (tran == NEWO_TRANS) {
                   time_ms = 20;
                } else if (tran == PAYMENT_TRANS) {
                   time_ms = 13;
                } else if (tran == ORDER_STAT_TRANS) {
                   time_ms = 16;
                } else if (tran == STOCK_TRANS) {
                   time_ms = 16;
                } else if (tran == DELIVERY_TRANS) {
                   time_ms = 16;
                }
                time_ms += rt_increment;
                timeP->tv_sec = 0;
                timeP->tv_nsec = time_ms * 1000000;
  }
  return(timeP);
}

/**  A simulated new order transaction  */
void sim_new_order(dataP)
 newOrder_data_t *dataP;
{
 int i;
 static int next_id = 100;
 struct timespec wait_time;
 static int lasttime = 0;
 struct timeval now;
 static int periods = 0;

 get_local_time(&now);
 if (now.tv_sec - lasttime > period_to_check_tran) {
  static trans[3];  /*  Keep the counts for the last 5 periods  */
  lasttime = now.tv_sec;

  if ((trans[1] - trans[0] < 2) &&
                (trans[2] - trans[1] < 2)) {
    rt_increment = 0;
    periods = 0;
    more_srv_work = getenv("TPCC_MORE_SERVER_WORK") ?
                                        atoi(getenv("TPCC_MORE_SERVER_WORK")) : 0;
    err_printf("Nothing much happening - resetting test\n");
  } else {
    periods++;
    if (periods % (period_to_add_rt / period_to_check_tran) == 0) {
                rt_increment += more_srv_work;
                err_printf("rt_increment now %d\n", rt_increment);
    }
  }
  trans[0] = trans[1];
  trans[1] = trans[2];
  trans[2] = next_id;
 }


 if (null_with_sleep)
   pthread_delay_np(get_wait_time(&wait_time,NEWO_TRANS));

 mat_mult(num_mults);
 sprintf((char *)dataP->c_last, "BARBARBAR");
 sprintf((char *)dataP->c_credit, "GC");
 dataP->c_discount = 0.33;
 dataP->o_id = next_id++;
 sprintf((char *)dataP->entry_date, "17-12-1995.12:33:56");
 dataP->total = 99.1;
 dataP->w_tax = 0.729;
 dataP->d_tax = 0.15;
 for (i=0; i<dataP->o_ol_cnt; i++) {
   dataP->item[i].price = dataP->item[i].ol_i_id % 1000;
   sprintf((char *)dataP->item[i].name_i, "item %d", i);
   dataP->item[i].s_quantity = i;
   dataP->item[i].brand_generic[0] = i%2 ? 'O' : 'E',
   dataP->item[i].brand_generic[1] = '\0';
   dataP->item[i].ol_amount =
     dataP->item[i].price * dataP->item[i].ol_quantity;
  }

 if ((dataP->item[dataP->o_ol_cnt - 1].ol_i_id < 1) ||
     (dataP->item[dataP->o_ol_cnt - 1].ol_i_id > 100000)) {
   dataP->header.returncode = INVALID_NEWO;
 } else if (RANDOM() % 90 == 0) {
   dataP->header.returncode = SIM_ERROR_CODE;
 } else {
   dataP->header.returncode = TPCC_SUCCESS;
 }
 return;
}

/**  A simulated payment transaction */
void sim_payment(dataP)
    payment_data_t *dataP;
{
   struct timespec wait_time;

   if (null_with_sleep)
     pthread_delay_np(get_wait_time(&wait_time,PAYMENT_TRANS));
   mat_mult(num_mults);

   dataP->c_id = 1;
   dataP->c_credit_lim = 100.9;
   dataP->c_discount = 0.2;
   dataP->c_balance = 11.1;

   sprintf((char *)dataP->c_first, "%-16s", "c_first");
   sprintf((char *)dataP->c_middle, "%-2s", "MI");
   sprintf((char *)dataP->c_last, "%-16s", "c_last");
   sprintf((char *)dataP->c_street_1, "%-20s", "c_street_1");
   sprintf((char *)dataP->c_street_2, "%-20s", "c_street_2");
   sprintf((char *)dataP->c_city, "%-20s", "c_city");
   sprintf((char *)dataP->c_state, "%-2s", "PA");
   sprintf((char *)dataP->c_zip, "%-9s", "152111111");
   sprintf((char *)dataP->c_phone, "%-16s", "6522573904218222");
   sprintf((char *)dataP->c_date, "%-19s", "28-11-1995");
   sprintf((char *)dataP->c_credit, "%-2s", "GC");
   sprintf((char *)dataP->pay_date, "%-19s", "17-12-1995.12:39:13");
   sprintf((char *)dataP->d_street_1, "%-20s", "d_street_1");
   sprintf((char *)dataP->d_street_2, "%-20s", "d_street_2");
   sprintf((char *)dataP->d_city, "%-20s", "d_city");
   sprintf((char *)dataP->d_state, "%-2s", "PA");
   sprintf((char *)dataP->d_zip, "%-9s", "152111111");
   sprintf((char *)dataP->w_street_1, "%-20s", "w_street_1");
   sprintf((char *)dataP->w_street_2, "%-20s", "w_street_2");
   sprintf((char *)dataP->w_city, "%-20s", "w_city");
   sprintf((char *)dataP->w_state, "%-2s", "OH");
   sprintf((char *)dataP->w_zip, "%-9s", "142411111");

   if (RANDOM() % 70 == 0) {
      dataP->header.returncode = SIM_ERROR_CODE;
   } else {
      dataP->header.returncode = TPCC_SUCCESS;
   }
}

/**  A simulated stock level  transaction */
void sim_stock_level(dataP)
    stockLevel_data_t *dataP;
{
   struct timespec wait_time;

   if (null_with_sleep)
     pthread_delay_np(get_wait_time(&wait_time,STOCK_TRANS));

   mat_mult(num_mults);

   dataP->stock_count = 12;
   if (RANDOM() % 80 == 0) {
                dataP->header.returncode = SIM_ERROR_CODE;
   } else {
                dataP->header.returncode = TPCC_SUCCESS;
   }
}


/**  A simulated delivery transaction */
void sim_delivery(dataP)
    delivery_data_t *dataP;
{
   struct timespec wait_time;

   if (null_with_sleep)
     pthread_delay_np(get_wait_time(&wait_time,DELIVERY_TRANS));

   dataP->start_queue = 2.2;
   dataP->header.returncode = TPCC_SUCCESS;
}

/**  A simulated order status transaction  */
void sim_order_status(dataP)
    orderStatus_data_t *dataP;
{
   int i;
   struct timespec wait_time;

   if (null_with_sleep)
```

```
    pthread_delay_np(get_wait_time(&wait_time,ORDER_STAT_TRANS));

  mat_mult(num_mults);


  dataP->c_id = dataP->c_id ? dataP->c_id : 99;
  strcpy((char *)dataP->c_first, "Jerome");
  strcpy((char *)dataP->c_middle, "LB");
  strcpy((char *)dataP->c_last, "Trevoe");
  dataP->c_balance = 90.78;
  dataP->o_id = 99;
  strcpy((char *)dataP->entry_date, "06-12-1995.16:42:28");
  dataP->o_carrier_id = 9;
  dataP->o_ol_cnt = 7;

  for (i=0; i<dataP->o_ol_cnt; i++) {
                dataP->item[i].ol_supply_w_id= 1;
                dataP->item[i].ol_i_id= dataP->w_id * 10 + dataP->d_id;
                dataP->item[i].ol_quantity= 10 * (i+1);
                dataP->item[i].ol_amount = dataP->item[i].ol_quantity * 10.1;
                strcpy((char *)dataP->item[i].delivery_date,"NOT DELIVR");
  }


if (RANDOM() % 90 == 0) {
  dataP->header.returncode = SIM_ERROR_CODE;
} else {
  dataP->header.returncode = 0;
 }
}

/*
*  mat_mult
*               Multiply the above two matrices
*/
static void mat_mult(iter)
   int iter;
{
 float res[ROWS][COLS];
 int i, j, k;
 int a_num_rows = ROWS;
 int a_num_columns= COLS;
 int b_num_rows = ROWS;
 int b_num_columns= COLS;

 for (; iter>0; iter--) {
  for (i=0; i<a_num_rows; i++) {
    for (j=0; j<b_num_columns;j++) {
                res[i][j] = 0;
                for (k=0; k<b_num_rows; k++) {
                  res[i][j] += matrix_a[i][k] * matrix_b[k][j];
                }
    matrix_a[i][j]= res[i][0];
    }
  }
  pthread_yield();
 }
}
```

server.h

```
/*
*               server.h
*
* $Revision: 1.11 $
* $Date: 1999/05/06 21:28:31 $
* $Log:             $
*
*
* $TALog: server.h,v $
* Revision 1.11  1999/05/06  21:28:31  oz
* - Removed all the .. from the includes
* - Added -I.. to the  makefiles instead
* - Moved all the thread related code and connection
*   selection to serverMon.c
* [from r1.9 by delta oz-24309-TPCC-add-oracle8.1-code, r1.5]
*
* Revision 1.9  1999/01/12  20:52:59  wenjian
* Define MAPOBJNAMEFORMAT so that the server processes and dll can communicate
* via the shared file mappings.
* [from r1.8 by delta wenjian-23856-TPCC-integrate-with-NT-performance-monitor,r1.1]
*
* Revision 1.8  1998/12/14  20:27:57  wenjian
* Made corresponding changes due to data structure change of tran_info_t.
*
* - change server_tran_t
* [from r1.7 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.3]
*
* Revision 1.7  1998/12/11  16:14:20  wenjian
* Add code for checking statistic data in a single variable and collecting
* statistic data based on iStatsFrequency.
*
* - Add server_tran_t and server_info_t
* [from r1.6 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.1]
*
* Revision 1.6  1998/11/09  16:59:48  wenjian
* In this revision, most of the changes are related to the directory of header
* files after directory reorganization. Other changes include adding or removing
* files to put them in the right directories. Makefiles are written for NT
* platform so that nmake is working on NT now. Need a top level Makefile for all
* the directories.
* [from r1.5 by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.2]
*
* Revision 1.5  1998/11/09  14:48:25  wenjian
* In an effort to make a new directory structure for TPCC, this delta
* creates two directories: tpcc/client and tpcc/server. All the files
* for this revision are copied from tpcc/sp-tpcc without any change.
* Further change may be needed for some files due to the change of
* the directory structure.
* [added by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.1]
*
* Revision 1.9  1998/10/22  15:33:05  wenjian
* Make changes to Encina server code to connect with SQL server and add
* callsql.c and sql directory.
*
* Add delivery_sql_t to deal with SYSTEMTIME struct used in SQL
* [from r1.7 by delta wenjian-23529-TPCC-integrate-with-SQL-server, r1.1]
*
* Revision 1.7  1998/01/23  15:08:50  oz
* - Updated the SP TPCC directory to the latest files used
*   during the SP tpcc audit.
* [from r1.6 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
*
*
 */

/**  server.h  **/

/**  Declarations common to all the server modules  **/


#ifndef TPCC_SERVER_H
#define TPCC_SERVER_H

#include "common/tpcc_type.h"

#define get_dbname_from_id(i) rmList[i].dbName
#define MAPOBJNAMEFORMAT "srv_%s_PA%d"

#ifdef WIN32
typedef struct {
    delivery_data_t  data;
    SYSTEMTIME       queue_time;
} delivery_sql_t;
#endif

typedef enum {
    mon_server = 11
} server_type_t;

typedef struct {
    int num;
    double RTtotal;
    int RTcount;
} server_tran_t;

typedef struct {
    server_tran_t tran[MAX_TRAN_TYPE + 1];
    int    total_trans;
} server_info_t;

extern int server_no_db;
extern int serverIdNumber;
extern int server_init;
extern server_type_t server_type;
extern int get_db_for_wh(int);

#endif   /*  TPCC_SERVER_H  */
```

serverDebug.h

```
/*
 * serverDebug.h
 *
 * $Revision: 1.5 $
 * $Date: 1998/11/09 14:48:25 $
 * $Log:          serverDebug.h,v $
 * Revision 4.4  95/05/16  10:55:40  10:55:40  tpcc (TPCC Benchmark)
 * Added necessary RCS ident strings
 *
 *
 */
#ifndef SERVER_DEBUG
#define SERVER_DEBUG

#include <utils/trace.h>
```

```
#ifdef DEFINE_SERVER_DEBUG
long serverDebug = 0;
#else
extern long serverDebug;
#endif

#ifdef TRACE_TRANS
#define TRACETRAN(list) logprintf list
#else
#define TRACETRAN(list)
#endif

#ifdef DEBUG_SERVER
#define AUDITLOG(list) if (serverDebug & AUDIT_TRANS) UNCOND_EVENT list
#define NEWOLOG(list) if (serverDebug & DBG_NEWO) err_printf list
#define PAYLOG(list) if (serverDebug & DBG_PAY) err_printf list
#define OSLOG(list) if (serverDebug & DBG_OS) err_printf list
#define STKLOG(list) if (serverDebug & DBG_STK) err_printf list
#define DEBUGP(list) if (serverDebug) err_printf list
#else
#define AUDITLOG(list)
#define NEWOLOG(list)
#define PAYLOG(list)
#define OSLOG(list)
#define STKLOG(list)
#define DELLOG(list)
#define DEBUGP(list)
#endif

#define ERRLOG(list) err_printf list
#define SQL_RET_CODE(var, code) var = (code)

/* Fix DPRINT to write on a debugging unit that can get set differently
for delivery */

#ifdef UNIT_TEST
#define DPRINT(list) dprint list
#define DELPRINT(list) delprint list
#else
#define DPRINT(list)
#define DELPRINT(list)
#endif

#define DBG_NEWO        0x0001
#define DBG_PAY                 0x0002
#define DBG_OS                  0x0004
#define DBG_STK                 0x0008
#define DBG_DEL                 0x0010
#define DBG_ERR                 0x0020
#define AUDIT_TRANS     0x0100

#endif                  /* SERVER_DEBUG */
```

serverMon.c

```
/*
* serverMon.c
*
* $Revision: 1.23 $
* $Date: 1999/05/28 19:44:17 $
* $Log:         serverEncina.c,v $
*
* $TALog: serverMon.c,v $
* Revision 1.23  1999/05/28  19:44:17  wenjian
* Add create_null_connection and clean_null_connection so that
* we can run with NULL DB.
* [from r1.22 by delta wenjian-24433-TPCC-clean-up-and-update, r1.4]
*
* Revision 1.22  1999/05/28  14:30:22  wenjian
* - Fix a bug for calling get_thread_data()
* [from r1.21 by delta wenjian-24433-TPCC-clean-up-and-update, r1.3]
*
* Revision 1.21  1999/05/26  16:29:59  wenjian
* Sync with Austin code, and sync code for Oracle DB and SQL server.
*
* - Fix some minor bugs
* [from r1.20 by delta wenjian-24433-TPCC-clean-up-and-update, r1.2]
*
* Revision 1.20  1999/05/06  21:28:31  oz
* - Removed all the .. from the includes
* - Added -I.. to the  makefiles instead
* - Moved all the thread related code and connection
*   selection to serverMon.c
*
* - Added connection and thread related code.
* - pre_DB and post_DB always get called.
* - get_db_ready has only 2 parameters
* - added tran_info_t, total_tran_count_t, and thread_info_t
* - Added clean_thread_data and get_thread_data
* - Preallocate all the connections if necessary after
*   initializing the DB.
* - A connection is created by calling the DB specific
*   create_connection() which returns a void* handle to
*   be passed to all the transactions.
* [from r1.18 by delta oz-24309-TPCC-add-oracle8.1-code, r1.5]
*
* Revision 1.18  1999/04/20  15:11:28  oz
* [merge of changes from 1.13 to 1.17 into 1.12]
*
* Revision 1.17  1999/04/19  20:14:49  oz
* - Moved all the simulated code to server.c
* - Created nulldb.c for compilation with no DB
* [from r1.13 by delta oz-24331-TPCC-move-sim-code-to-common-file, r1.1]
*
* Revision 1.12  1999/01/12  20:53:00  wenjian
* - Call initialization function perfSrvDataInit to create the shared file
*   mapping for this server process
* - Change server_info to pServerInfo
* [from r1.11 by delta wenjian-23856-TPCC-integrate-with-NT-performance-monitor, r1.1]
*
* Revision 1.11  1998/12/14  20:27:58  wenjian
* Made corresponding changes due to data structure change of tran_info_t.
*
* - Made changes for server_tran_t
* - Add tran_type to FUNCTION_BEGIN, FUNCTION_END, pre_DB, and post_DB
* [from r1.10 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.3]
*
* Revision 1.10  1998/12/11  16:14:20  wenjian
* Add code for checking statistic data in a single variable and collecting
* statistic data based on iStatsFrequency.
*
* - Change pre_oracle to pre_DB, post_oracle to post_DB
* - Add code to collect server RT in a global var
* [from r1.9 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.1]
*
* Revision 1.9  1998/12/08  18:55:22  wenjian
* - Remove "statsFrequency=" command line argument.
* - Check rpc header for stats
* [from r1.8 by delta wenjian-23785-TPCC-pass-statsFrequency-from-client-to-server, r1.1]
*
* Revision 1.8  1998/12/07  20:04:16  wenjian
* Remove interfaces for explicit bindings
* [from r1.7 by delta wenjian-23742-TPCC-update-with-Raliegh-code, r1.2]
*
* Revision 1.7  1998/11/24  21:46:03  wenjian
* - Remove COLLECT_TIMESTAMPS; use command line argument iStatsFrequency
*   instead
* - Take care of MULTIPLE_INTERFACE and SINGLE_INTERFACE
* [from r1.6 by delta wenjian-23742-TPCC-update-with-Raliegh-code, r1.1]
*
* Revision 1.6  1998/11/09  16:59:48  wenjian
* In this revision, most of the changes are related to the directory of header
* files after directory reorganization. Other changes include adding or removing
* files to put them in the right directories. Makefiles are written for NT
* platform so that nmake is working on NT now. Need a top level Makefile for all
* the directories.
* [from r1.5 by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.2]
*
* Revision 1.5  1998/11/09  14:48:25  wenjian
* In an effort to make a new directory structure for TPCC, this delta
* creates two directories: tpcc/client and tpcc/server. All the files
* for this revision are copied from tpcc/sp-tpcc without any change.
* Further change may be needed for some files due to the change of
* the directory structure.
* [added by delta wenjian-23677-TPCC-reorganize-directory-structure, r1.1]
*
* Revision 1.36  1998/11/06  16:10:56  wenjian
* - Increase the range for the number of threads communicating to the DB.
* - Print warning if the number of threads is out of the range.
* [from r1.35 by delta wenjian-23646-TPCC-clean-up-source-code, r1.1]
*
* Revision 1.35  1998/10/22  21:30:47  wenjian
* [merge of changes from 1.20 to 1.29 into 1.34]
*
* Revision 1.29  1998/10/08  14:18:02  dongfeng
* Add codes for doing web-based tpcc.
* [from r1.20 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.1]
*
* Revision 1.34  1998/10/22  19:43:38  wenjian
* [merge of changes from 1.28 to 1.32 into 1.30]
*
* Revision 1.32  1998/10/22  16:25:13  wenjian
* Multi-threaded version.
*
* - Open server_logtrans for err_printf() and deliLog for delivery server
* - Call get_time_init() for get_local_time
* - Changes for multi-threaded version
* [from r1.31 by delta wenjian-23529-TPCC-integrate-with-SQL-server, r1.2]
*
* Revision 1.31  1998/10/22  15:33:05  wenjian
* Make changes to Encina server code to connect with SQL server and add
* callsql.c and sql directory.
*
* - Add SYSTEMTIME *queue_time to deferred_dvry_t for NT
* - Allocate space and set value for queue_time
* - Pass delivery_sql_t instead of delivery_data_t for SQLdel on NT
* [from r1.28 by delta wenjian-23529-TPCC-integrate-with-SQL-server, r1.1]
*
* Revision 1.30  1998/10/08  18:03:03  gerstl
* Changes to allow configurations where some servers only service
* specific transaction types. Split transaction interfaces by type.
*
* Transaction interface support is based upon a bitmap passed to the
* server.
* [from r1.28 by delta gerstl-23515-TPCC-allow-separate-online-transaction-interfaces, r1.1]
```

```
*
* Revision 1.28  1998/10/07  15:49:49  gerstl
* [merge of changes from 1.22 to 1.26 into 1.27]
*
* Revision 1.26  1998/09/03  20:22:02  wenjian
* Sync with Austin code: mostly use servrMon.c in Austin.
* [from r1.25 by delta wenjian-23183-TPCC-sync-AIX-code-with-Austin,r1.4]
*
* Revision 1.25  1998/09/03  16:07:12  wenjian
* Remove UNCOND_EVENT which is not in austin code.
* [from r1.24 by delta wenjian-23183-TPCC-sync-AIX-code-with-Austin,r1.3]
*
* Revision 1.24  1998/09/02  15:43:30  wenjian
* Define num_worker_threads.
* [from r1.23 by delta wenjian-23183-TPCC-sync-AIX-code-with-Austin,r1.2]
*
* Revision 1.23  1998/08/28  18:30:02  wenjian
* This delta sync the TPCC code with Austin.
*
* - Take care of 1 thread per PA
* - Update with Austin code
* - Remove the old code starting from #ifdef THIS_WAS_THE_OLD_CODE
* - Remove imp TPCCDvryInfo()
* [from r1.22 by delta wenjian-23183-TPCC-sync-AIX-code-with-Austin,r1.1]
*
* Revision 1.27  1998/09/26  10:27:33  oz
* Changes for NT.
* [from r1.22 by delta oz-23339-TPCC-update-for-NT, r1.1]
*
* Revision 1.22  1998/08/18  14:38:43  wenjian
* Minor change
* [from r1.20 by delta wenjian-21750-TPCC-changes-for-porting-on-NT,r1.4]
*
* Revision 1.20  1998/02/17  22:07:06  wenjian
* Minor changes for NT
* [from r1.19 by delta wenjian-21750-TPCC-changes-for-porting-on-NT,r1.1]
*
* Revision 1.19  1998/01/30  15:12:28  oz
* - Remove the explicit binding functions from the idl
*   files and from serverMon.c
* [from r1.18 by delta oz-21697-TPCC-remove-explicit-binding-code,r1.2]
*
* Revision 1.18  1998/01/24  14:17:06  oz
* - User server name to identify server and name delivery file
* - Use env variable HOME instead of /home/encina if HOME is set
*
* - Removed the machine list
*   The server ID is computed from the first number found in the host name
* [from r1.17 by delta oz-21687-TPCC-use-server-name-to-identify-process,r1.1]
*
* Revision 1.17  1998/01/23  21:59:00  oz
* - In order to simplify the Encina TPCC code: Merge the four
*   online transactions into 1 interface
* - Moved all the scripts to a scripts subdirectory
* - Removed unused files
* [from r1.16 by delta oz-21671-TPCC-merge-online-transaction-interfaces,r1.1]
*
* Revision 1.16  1998/01/23  15:08:53  oz
* - Updated the SP TPCC directory to the latest files used
*   during the SP tpcc audit.
* [from r1.15 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
*
*
*                   serverMon.c
*
*  Code that is monitor specific.
*
*/

#include <sys/types.h>
#ifndef WIN32
#include <unistd.h>
#else
#include <process.h>
#include <winsock.h>
#endif
#include <stdio.h>
#include <stdarg.h>
#include <time.h>
#include <tmxa/tmxa.h>
#include <tc/tc.h>
#include <tpm/mon/mon_server.h>

#if defined (solaris)
#include <dce/pthread.h>
#else   /* solaris */
#include <pthread.h>
#endif  /* solaris */
#include <utils/trace.h>

#include "common/databuf.h"
#include "common/utilities.h"
#include "serverDebug.h"
#include "common/delivery.h"
#ifdef MULTIPLE_INTERFACE
#include "common/neworder.h"
#include "common/payment.h"
#include "common/stocklevel.h"
#include "common/orderstatus.h"
#else
#include "common/tpcc_trans.h"
#endif
#include "server.h"

#  define FUNCTION_BEGIN(name, dataP, tran_type, infoP)                 \
                pre_DB(name, &(dataP)->header, tran_type, infoP);

#  define FUNCTION_END(name, dataP, tran_type, infoP)  \
                post_DB(name, &(dataP)->header, tran_type, infoP);  \


#ifdef WIN32
#define CASECMP(x,y) _stricmp(x,y) == 0
#define getpid  _getpid
extern void TPCexit();
#else
#define CASECMP(x,y) strcasecmp(x,y) == 0
extern void TPCexit();
#endif

extern void *create_connection();
extern void clean_connection(void *);
extern int get_db_ready(char *, int);

inModule("serverMon");

static void start_deferred_delivery_threads();
static void queue_delivery(delivery_data_t*dataP);
static void *create_null_connection();
static void clean_null_connection(void*ptr);

extern int server_null_test;

extern void err_printf( char *format, ...);
extern int get_db_ready(char *, int);
extern void logprintf(char *format, ...);
extern server_info_t *perfSrvDataInit(char *, int);
extern void *start_bg_thread();

static void get_mon_server_env();

server_type_t server_type = mon_server;

char *tpcc_serverName = NULL;
char *dbName = NULL;
int total_num_warehouses;
int num_deferred_dvry_threads = 1;
int num_worker_threads = 1;
int dvry_queue_size = 3000;
server_info_t *pServerInfo = NULL;
char oracle_home[256];  /* will be used in tpccpl.c */

typedef struct {
  pthread_mutex_t       lock;
  pthread_cond_t        q_cond;
  pthread_cond_t        work_cond;

  int                                   num_waiters;  /*  Number of new requests waiting  */

  int                                   head, tail;
  int                                   allocated;  /* Total size of the queue  */
  int                                   size;   /* Num elements currently there  */
#ifdef WIN32
  SYSTEMTIME            *queue_time;
#endif
  delivery_data_t       *data;
} deferred_dvry_t;

static deferred_dvry_t deferred_dvry_data;
#define MAX_DVRY_QUEUE deferred_dvry_data.allocated

/*
*  Information about one transaction type
*/
typedef struct {
  int num;
  int errs;
  double RT;
} tran_info_t;

/*
* total_tran_count_t
*
* structure that holds the total count of transaction of each type
* as well as the reposne times.
*
*/
typedef struct {
  tran_info_t tran[MAX_TRAN_TYPE + 1];
  int errors;
} total_tran_count_t;



typedef struct {
  void *cnP;  /*  DB specific connection to be used by this thread */

  int calls;      /*  Number of times it was used  */
```

```
  int errors;     /*  Total number of errors on this connection  */
  int calls_last_err;  /*  Number of calls when the last error occured  */
  int consecutive_errs; /*  Number of consecutive errs  */
  int connect_time;          /*  Time (seconds) connections was created  */

  /*  For debug  */
  int state;                          /*  State of the connection  */
  struct timeval tran_time;     /*  Time this tran started */
  int cur_tran_type;
  void *cur_tran_dataP;
  total_tran_count_t stat;
  int printed;

} thread_info_t;


#define SVR_STATE_NONE                  0
#define SVR_STATE_SENT                  1
#define SVR_STATE_REPLIED          2
#define SVR_STATE_ERR              3

/*  Connection related data structures */

static void clean_thread_data(void *ptr);

pthread_key_t thread_key;
pthread_mutex_t init_lock;
thread_info_t *info_array = NULL;  /*  Array of thread data */
int num_threads = 0;  /*  number of threads that have already been init */
int next_thread = 0;  /*  next thread id: next entry in the array  */

int preallocate_cn = 1;  /*  Should all connections be preallocated */
int num_connections = 0;
int num_allocated = 0;

static thread_info_t *get_thread_data();

static void display_mon_env()
{
  char *env_str;
  char envMsg[64];

#define DISPLAY_ENV_VAR(var) \
    if ((env_str = getenv(var)) != NULL) { \
                UNCOND_EVENT("%s == '%s'\n", var, env_str); \
    } else { \
                UNCOND_EVENT("%s not set\n", var);  \
    }

  UNCOND_EVENT("TPCC Server display env.  ID: %d\n", serverIdNumber);

  /*
   *              For debugging purpose: have the first PA
   *  display the following information
   */


  if ((serverIdNumber & 0xff) == 0) {
    DISPLAY_ENV_VAR("RPC_SUPPORTED_PROTSEQS");
    DISPLAY_ENV_VAR("RPC_UNSUPPORTED_NETADDRS");
    DISPLAY_ENV_VAR("ENCINA_BINDING_TIMEOUT");
    DISPLAY_ENV_VAR("ENCINA_THREAD_POOL_QUEUE_LENGTH");
    DISPLAY_ENV_VAR("ENCINA_THREAD_POOL_QUEUE_LENGTH");
    DISPLAY_ENV_VAR("ENCINA_TPOOL_SIZE");
    DISPLAY_ENV_VAR("ENCINA_RPC_THREAD_STACK_SIZE");
  }
}



/*  get_server_index()  --       This is used for debug purposes only
*
*  Return the server index for this server.
*  By convention, all the client machines have similar
*  names with different numbers, as in client1 client2, ...
*  If the convention is followed the server index is the first
*  number found.  Otherwise, it is 0.
*
*/
static int get_server_index()
{
 int i, ind;
 char host_name[128];
 if (0 == gethostname(host_name, sizeof(host_name))) {
  err_printf("Machine is on host %s\n", host_name);
  ind = strcspn(host_name, "0123456789");
  return(atol(host_name + ind));
 }

 return(0);
}


static parse_cmd_line(int argc, char *argv[], char **scheduling, int *interface_type)
{
  int nextInd = 1;
  char usageStr[128];
  int envRetrieval;
  if ((nextInd + 3) > argc) {
      sprintf(usageStr,
          "Not enough parameters.  Usage: %s  [-no_db] interfaces schedulingPolicy envRetrievalFlag
[dvry=#] [debugFlag] [db:<rmName>] [cn=#].", argv[0]);

      fprintf(stderr, "%s\n", usageStr);
      mon_TerminateServer(usageStr);

  } else {
      if (strcmp(argv[nextInd], "-no_db") == 0) {
          server_null_test = 1;
          fprintf(stderr, " ---== NULL test ==---\n");
          nextInd++;
      }
      *interface_type = strtol(argv[nextInd++], NULL, 0);
      *scheduling = argv[nextInd++];
      envRetrieval = atoi(argv[nextInd++]);

      while (nextInd < argc) {
          if (strncmp(argv[nextInd], "db:", 3) == 0) {
                                                                  dbName = argv[nextInd] + 3;
              nextInd++;
          } else if (strncmp(argv[nextInd], "dvry=", 5) == 0) {
              num_deferred_dvry_threads = atol(argv[nextInd] + 5);
              if (num_deferred_dvry_threads < 0 || num_deferred_dvry_threads > 200) {
                                  err_printf("num_deferred_dvry_threads was %d (>200), reset to 10\n",
num_deferred_dvry_threads);
                  num_deferred_dvry_threads = 10;
                                  }
              nextInd++;
          } else if (strncmp(argv[nextInd], "dvryQ=", 6) == 0) {
              dvry_queue_size = atol(argv[nextInd] + 6);
              if (dvry_queue_size < 1 || dvry_queue_size > 200000)
                  dvry_queue_size = 10;
              nextInd++;
          } else {
              serverDebug = atol(argv[nextInd++]);
          }
      }
  }
}


static void set_scheduling(char *scheduling)
{
  mon_paAccess_t paAccess;
  UNCOND_EVENT("Setting Scheduling Policy: %s\n", scheduling);

  if(CASECMP(scheduling, "MON_CONCURRENT_SHARED")) {
      paAccess = MON_CONCURRENT_SHARED;
  } else if(CASECMP(scheduling, "MON_EXCLUSIVE")) {
      num_deferred_dvry_threads = 1;
      paAccess = MON_EXCLUSIVE;
  } else if(CASECMP(scheduling, "MON_SHARED")) {
                  num_deferred_dvry_threads = 1;
      paAccess = MON_SHARED;
  } else {
                  err_printf("Invalid Policy: '%s'\n", scheduling);
      mon_TerminateServer("Invalid scheduling policy specified.");
  }

  ENCINA_CALL("mon_SetSchedulingPolicy",
          mon_SetSchedulingPolicy(paAccess));
}

static void register_interfaces(int interface_type)
{
  extern FILE *deliLog;
  char *env_str;
  int env_val;

  UNCOND_EVENT("Registering interfaces\n");

  num_worker_threads = 0;

  /* interface_type is a bitmap of the interfaces this
   * server needs to support.
   */
#ifdef MULTIPLE_INTERFACE
  if (interface_type & NEWO_INTERFACE) {
                  ENCINA_CALL("mon_InitServerInterface",
                                  mon_InitServerInterface(MON_SERVER_INTERFACE(neworder,1,0)));
  }
  if (interface_type & PAYMENT_INTERFACE) {
                  ENCINA_CALL("mon_InitServerInterface",
                                  mon_InitServerInterface(MON_SERVER_INTERFACE(payment,1,0)));
  }
  if (interface_type & ORDER_STAT_INTERFACE) {
                  ENCINA_CALL("mon_InitServerInterface",
                                  mon_InitServerInterface(MON_SERVER_INTERFACE(orderstatus,1,0)));
  }
  if (interface_type & STOCK_INTERFACE) {
                  ENCINA_CALL("mon_InitServerInterface",
                                  mon_InitServerInterface(MON_SERVER_INTERFACE(stocklevel,1,0)));
  }
#else
  if (interface_type & ONLINE_INTERFACES) {
                  ENCINA_CALL("mon_InitServerInterface",
                                  mon_InitServerInterface(MON_SERVER_INTERFACE(tpccTrans,1,0)));
  }
```

```
#endif

  if (interface_type & DELIVERY_INTERFACE) {
#ifdef WIN32
    deliLog = fopen("deliLog.out", "w");
#endif
    if (num_deferred_dvry_threads > 0) {
      start_deferred_delivery_threads();
    }
    ENCINA_CALL("mon_InitServerInterface",
                                mon_InitServerInterface(MON_SERVER_INTERFACE(delivery,1,0)));
  } else {
    num_deferred_dvry_threads = 0;
  }

  /* ENCINA_TPOOL_SIZE and ENCINA_APPL_TPOOL_SIZE
   * are set in tpccCommon.tcl for each
   * server started. If we are delivery only, we don't care
   * about it, otherise we need to adjust num_worker_threads
   */
  if (interface_type & ONLINE_INTERFACES) {
    if ((env_str = getenv("ENCINA_APPL_TPOOL_SIZE")) != NULL) {
      env_val = atol(env_str);
      if (env_val >= 0 && env_val < 1000)
        num_worker_threads += env_val;
      else {
                  err_printf("ENCINA_APPL_TPOOL_SIZE was %d, reset to 10\n", env_val);
        num_worker_threads += 10;
                }
    }
    if ((env_str = getenv("ENCINA_TPOOL_SIZE")) != NULL) {
      env_val = atol(env_str);
      if (env_val >= 0 && env_val < 1000)
        num_worker_threads += env_val;
      else {
                  err_printf("ENCINA_TPOOL_SIZE was %d, reset to 10\n", env_val);
        num_worker_threads += 5;
                }
    }
    if (num_worker_threads < 1) num_worker_threads = 1;
  }

}

void main(argc,argv)
  int argc;
  char *argv[];
{
  extern FILE *server_logtrans;
  int rc;
  int pa_num;
  char *scheduling = "";
  int rmId;
  char intermediary[256];
  extern int serverPid;
  int interface_type = ALL_INTERFACE;
  int status;

  inFunction("server_Init");

  /* hard code first for a quick test */
  /* getenv didn't work, though we have ORACLE_HOME defined */
  /*    strcpy(oracle_home,getenv("ORACLE_HOME")); */
  strcpy(oracle_home, "/home/oracle");

  server_logtrans = fopen("server_print.out", "w");

  get_time_init();

  serverPid = getpid();
  UNCOND_EVENT("TPCC Server Starting\n");

  /*  Use the top 8 bits of the serverIdNumber to store the server index */
  serverIdNumber = (get_server_index() & 0xff) * 1000;

  parse_cmd_line(argc, argv, &scheduling, &interface_type);

  display_mon_env();

  DEBUGP(("Debug level set at %d\n", serverDebug));


  DEBUGP(("Creating thread data key"));
  if(status = pthread_keycreate(&thread_key, clean_thread_data)) {
                fprintf(stderr, "init_global_data: pthread_keycreate failed: %d\n", status);
                mon_TerminateServer("Cannot create a key for the thread data");
  }

  mon_RetrieveEnable(FALSE);

  err_printf("Setting scheduling %s.\n", scheduling);
  set_scheduling(scheduling);
  err_printf(" Registering interfaces \n");
  register_interfaces(interface_type);

  err_printf("Calling mon_init\n");
  ENCINA_CALL("mon_InitServer", mon_InitServer());
  ENCINA_CALL("mon_SetHandleCacheRefreshInterval",
          mon_SetHandleCacheRefreshInterval(300));
  pa_num = mon_RetrievePaNum();
  tpcc_serverName = mon_RetrieveServerId();
  if (pa_num > 0)
                serverIdNumber += pa_num;
  err_printf("PA Number %d, serverId %d (%s)\n",
                        pa_num, serverIdNumber, tpcc_serverName);


  num_connections = num_deferred_dvry_threads + num_worker_threads;

  if ((rc = get_db_ready(dbName, 0)) != 0) {
                char msg[128];
                sprintf(msg, "failed to open database tpcc/tpcc: %d", rc);
    WARNING("%s\n", msg);
    err_printf("%s\n", msg);
                mon_TerminateServer(msg);
  }
  if (preallocate_cn || num_connections == 1) {
                int i;
                thread_info_t *curP;
                /*  Preallocate all the desired connections  */
                logprintf("Preallocating %d connections to the DB\n", num_connections);
                info_array = (thread_info_t*)calloc(num_connections,sizeof(*info_array));

                for (i=0, curP = info_array; i<num_connections; i++, curP++) {
                  if (server_null_test)
                                curP->cnP = create_null_connection();
                  else
                                curP->cnP = create_connection();
                }
                num_allocated = num_connections;
  }

  /* initialize pServerInfo */
#ifdef WIN32
  pServerInfo = perfSrvDataInit(tpcc_serverName, pa_num);
#endif
  if (pServerInfo == NULL)
                pServerInfo = malloc(sizeof(server_info_t));
  memset(pServerInfo,0,sizeof(server_info_t));
#ifndef WIN32
  start_bg_thread();
#endif

  err_printf(">> Calling mon_BeginService()\n");

  ENCINA_CALL("mon_BeginService",mon_BeginService());

  fprintf(stderr, "mon_BeginService returned ... terminating\n");
  TPCexit();
}

static void clean_thread_data(void *ptr) {
  thread_info_t *threadP = (thread_info_t *)ptr;
  if (server_null_test)
                clean_null_connection(threadP->cnP);
  else
                clean_connection(threadP->cnP);
  err_printf("Closing connection 0x%p.  Called %d, %d errors\n",
                      threadP->cnP, threadP->calls, threadP->errors);
}

/*
*  The routine executed by the deferred delivery thread
*
*  Logic:
*                Wait until there is a valid request in the deferred delivery data.
*                After processing the request data_valid is set to FALSE
*                (allowing new requests to be queued).
*  This is a simple fixed size queue implemented in a cyclic array
*/
static void deferred_delivery()
{
  thread_info_t *infoP;
  pthread_mutex_lock(&deferred_dvry_data.lock);

  while (1) {
                if (deferred_dvry_data.size > 0) {
                  /*
                   *  There is a request to be processed
                   */
                  int ind = deferred_dvry_data.head % MAX_DVRY_QUEUE;
#ifdef WIN32
                  delivery_sql_t dbData;
                  dbData.data = deferred_dvry_data.data[ind];
                  dbData.queue_time = deferred_dvry_data.queue_time[ind];
#else
                  delivery_data_t data = deferred_dvry_data.data[ind];
#endif

                  deferred_dvry_data.head ++;
                  deferred_dvry_data.size --;

                  if (deferred_dvry_data.num_waiters > 0)
                                pthread_cond_signal(&deferred_dvry_data.q_cond);

                  if (deferred_dvry_data.head % 1000 == 0) {
                                err_printf("Processed %d deferred deliveries so far, queue size %d\n",
```

```
                                        deferred_dvry_data.head,
                                        deferred_dvry_data.size);
                }
                if (deferred_dvry_data.head > deferred_dvry_data.tail) {
                        err_printf("Error: Deferred Queue: head %d > tail %d\n",
                                        deferred_dvry_data.head,
                                        deferred_dvry_data.tail);
                        continue;
                }
                pthread_mutex_unlock(&deferred_dvry_data.lock);

#ifdef WIN32
                if (server_null_test) {
                        sim_delivery(&dbData);
                } else {
                        infoP = get_thread_data();
                        do_delivery(infoP->cnP,&dbData);
                }
#else
                if (server_null_test) {
                        sim_delivery(&data);
                } else {
                        infoP = get_thread_data();
                        do_delivery(infoP->cnP,&data);
                }
#endif

                DPRINT(("Deferred: Locking\n"));
                pthread_mutex_lock(&deferred_dvry_data.lock);

              } else {
                /*
                 *  Wait for a request to be queued
                 */
                DPRINT(("Deferred delivery waiting\n"));
                pthread_cond_wait(&deferred_dvry_data.work_cond,
                                        &deferred_dvry_data.lock);
        DPRINT(("Deferred: Awake\n"));
    }
  }
}

/*
*  queue_delivery
*
*  Queue a delivery request to be processed in the background
*  The queue is implemented as a simple queue of size 1.
*  if data_valid is true: there is already a request waiting in the queue
*  Sleep on a condition variable until the queue is empty.
*  Once the queue is empty put the request in the queue, wake up the
*  background thread and leave.
*/

static void queue_delivery(dataP)
 delivery_data_t *dataP;
{
  struct timeval now;
  int waited = 0;
  static int last_report_time = 0;
#ifdef WIN32
  SYSTEMTIME  queue_time;
#endif

  DPRINT(("queue: Locking\n"));
  pthread_mutex_lock(&deferred_dvry_data.lock);
  DPRINT(("queue: Locked\n"));

  while (deferred_dvry_data.size >= MAX_DVRY_QUEUE) {
                /*  The request queue is full
                 *  Wait until a request is processed and removed from the queue.
                 */
                deferred_dvry_data.num_waiters++;
                DPRINT((">> queue_delivery: %d waiters, size %d\n",
                        deferred_dvry_data.num_waiters, deferred_dvry_data.size));
                DPRINT(("Queue Delivery waiting, %d waiters\n",
                        deferred_dvry_data.num_waiters));
                pthread_cond_wait(&deferred_dvry_data.q_cond,
                                        &deferred_dvry_data.lock);
                deferred_dvry_data.num_waiters--;
                waited ++;
  }
  DPRINT(("Queuing delivery\n"));
  /*
   *  There is room in the queue.
   *  Enter the request and wake up the background thread
   */
#ifdef WIN32
  GetLocalTime(&queue_time);
  deferred_dvry_data.size++;
  deferred_dvry_data.data[deferred_dvry_data.tail % MAX_DVRY_QUEUE] = *dataP;
  deferred_dvry_data.queue_time[deferred_dvry_data.tail % MAX_DVRY_QUEUE] = queue_time;

#else
  get_local_time(&now);
  dataP->start_queue = (double)now.tv_sec + (now.tv_usec / 1000000.0);
  deferred_dvry_data.size++;
  deferred_dvry_data.data[deferred_dvry_data.tail % MAX_DVRY_QUEUE] = *dataP;
  if (now.tv_sec - last_report_time > 29) {
                err_printf("queue_delivery- %d waiters, size %d\n",
                        deferred_dvry_data.num_waiters, deferred_dvry_data.size);
                last_report_time = now.tv_sec;
  }
#endif

  deferred_dvry_data.tail++;
  DPRINT(("queue_delivery: Signalling\n"));
  pthread_cond_signal(&deferred_dvry_data.work_cond);

  DPRINT(("queue_delivery: Unlocking.  Tail %d, size %d Max %d\n",
                deferred_dvry_data.tail,
                deferred_dvry_data.size,
                MAX_DVRY_QUEUE));
  pthread_mutex_unlock(&deferred_dvry_data.lock);

  if (waited) err_printf(">> queue_delivery waited %d times\n", waited);
  dataP->header.returncode = TPCC_SUCCESS;
}

/*
* start_deferred_delivery_threads
*
*               Initialize the deferred delivery data structure and start
*               a background thread to process the delivery requests
*/
static void start_deferred_delivery_threads()
{
  pthread_t thread;
  int i;
  int rc;

  pthread_mutex_init(&deferred_dvry_data.lock, pthread_mutexattr_default);
  pthread_cond_init(&deferred_dvry_data.work_cond, pthread_condattr_default);
  pthread_cond_init(&deferred_dvry_data.q_cond, pthread_condattr_default);
  deferred_dvry_data.num_waiters = 0;
  deferred_dvry_data.head = 0;
  deferred_dvry_data.tail = 0;
  deferred_dvry_data.size = 0;
  deferred_dvry_data.allocated = dvry_queue_size;
  deferred_dvry_data.data =
                (delivery_data_t *)malloc(dvry_queue_size * sizeof(delivery_data_t));
#ifdef WIN32
  deferred_dvry_data.queue_time =
                (SYSTEMTIME *)malloc(dvry_queue_size * sizeof(SYSTEMTIME));
#endif

  /*
   *  Create the background delivery thread.
   */
  err_printf("Starting %d deferred delivery threads, queue size %d\n",
                        num_deferred_dvry_threads,
                        dvry_queue_size);
  for (i=0; i<num_deferred_dvry_threads;i++) {
     if ((rc = pthread_create(&thread,
                                                        pthread_attr_default,
                                                        (pthread_startroutine_t)deferred_delivery,
                                                        (pthread_addr_t)0)) != 0) {
                        WARNING("Failed to create delivery thread rc=%d\n", rc);
                        exit(1);
     }
     (void)pthread_detach(&thread);
  }
}

void exit_program(code)
   int code;
{
 char errMsg[55];
 sprintf(errMsg, "exit_program called with code %d", code);
 fprintf(stderr, "%s\n", errMsg);

 TPCexit();

 mon_TerminateServer(errMsg);
}

static char *thread_state_to_str(int state)
{
  char *retval;
  switch(state) {
                case SVR_STATE_NONE: retval = "None"; break;
                case SVR_STATE_SENT: retval = "Sent"; break;
                case SVR_STATE_REPLIED: retval = "Replied"; break;
                case SVR_STATE_ERR: retval = "Err"; break;
                default: retval = "unknown"; break;
  }
  return retval;
}


static thread_info_t *get_thread_data() {
  thread_info_t *dataP;
  struct timeval cur_time;

  /*  Get a thread structure.
   *  Each thread always uses the same connection.
   *  The first time the thread tries to talk to the DB it creates
   *  a connection, initializes it and stores it in a thread global
   *  data structure.
```

```
 *
 *  There is a special case for the single connection case:  If there
 *  is exactly one connection then it is global and not per thread.
 *  There may be maany threads but it is assumed that the application is
 *  responsible for synchronizing the threads so that no two threads
 *  ever use the connection at the same time.
 */
 if (num_connections == 1) {
                dataP = &info_array[0];
 } else {
    pthread_getspecific(thread_key, (pthread_addr_t *)&dataP);
 }
 if (dataP == NULL) {  /* No connection assigned to this thread */
                pthread_mutex_lock(&init_lock); /*  Initialize a connection*/
                get_local_time(&cur_time);

    fprintf(stderr, "get_cn> initializing threadslot\n");

                if (preallocate_cn) {
                   if (next_thread >= num_allocated) {
                                fprintf(stderr, "Too many threads, not enough connections\n");
                                mon_TerminateServer("Too many threads, not enough connections");
                   }
                   dataP = &info_array[next_thread++];
                } else {
                   dataP = (thread_info_t *)malloc(sizeof(thread_info_t));
                   memset(dataP, (char)0, sizeof(*dataP));
                   if (server_null_test)
                                dataP->cnP = create_null_connection();
                   else
                                dataP->cnP = create_connection();
                }
    pthread_setspecific(thread_key, dataP);  /*  Store it  */

    fprintf(stderr, "get_cn> initialized connection0x%x\n", dataP);
                pthread_mutex_unlock(&init_lock);

 }
 return dataP;
}


static void pre_DB(char *name, data_header *headerP,
                                int tran_type, thread_info_t *infoP)
{
 struct timeval tp;
 DPRINT(("> %s", name));
 get_local_time(&tp);
if (infoP != NULL) {
 infoP->cur_tran_type = tran_type;
 infoP->calls++;
 infoP->state = SVR_STATE_SENT;
 infoP->tran_time = tp;
}

 headerP->start_time.sec = tp.tv_sec;
 headerP->start_time.usec = tp.tv_usec;
}

static void post_DB(char *name, data_header *headerP,
                                 int tran_type, thread_info_t *infoP)
{
 struct timeval tp;
 DPRINT(("< %s\n", name));
 get_local_time(&tp);
 headerP->end_time.sec = tp.tv_sec;
 headerP->end_time.usec = tp.tv_usec;
 headerP->dtype = serverIdNumber;

if (infoP != NULL){
 infoP->tran_time = tp;
 infoP->state = SVR_STATE_REPLIED;
}

 pServerInfo->tran[tran_type].num++;
 /* store the RT info for this server */
 if (tran_type <= MAX_TRAN_TYPE && tran_type > 0) {
      pServerInfo->tran[tran_type].RTtotal +=
time_diff_ms(&(headerP->end_time),&(headerP->start_time));
      pServerInfo->tran[tran_type].RTcount ++;
 }

}

/*
* ----------------------  The following are the entry points
*                                                 for the RPCs arriving at the Server
*/

void impTPCCDbInfo(dataP,trpcStatus)
 dbInfo_data_t *dataP;
 trpc_status_t  *trpcStatus;
{
  UNCOND_EVENT("> impTPCCDbInfo");
  err_printf("> impTPCCDbInfo serverIdNumber=%d\n",serverIdNumber);
  dataP->server_id = serverIdNumber;
  err_printf("< impTPCCDbInfo\n");
}


void impTPCCNOInfo(h,dataP,trpcStatus)
 encina_handle_t h;
 dbInfo_data_t *dataP;
 trpc_status_t  *trpcStatus;
{
  impTPCCDbInfo(dataP,trpcStatus);
}


void impTPCCNewOrder(h,dataP,trpcStatus)
 encina_handle_t h;
 newOrder_data_t *dataP;
 trpc_status_t  *trpcStatus;
{

  static int numCalls = 0;
  thread_info_t *infoP = get_thread_data();
  FUNCTION_BEGIN("NewOrder", dataP, NEWO_TRANS, infoP);
  if (server_null_test) {
                sim_new_order(dataP);
  } else {
                do_new_order(infoP->cnP, dataP);
  }

  if ((dataP->header.returncode != TPCC_SUCCESS) &&
                (dataP->header.returncode != INVALID_NEWO)) {
    logprintf("< impTPCCNewOrder; rc=%d, sql=%d, isam=%d\n",
                                dataP->header.returncode,
                                dataP->header.sql_code,
                                dataP->header.isam_code);
  } else if (dataP->header.returncode == INVALID_NEWO) {
                DPRINT(("< impTPCCNewOrder INVALID_NEWO\n"));
  }
  if (++numCalls % 10000 == 0) {
                err_printf("impTPCCNewOrder so far %d\n", numCalls);
  }
  FUNCTION_END("NewOrder", dataP, NEWO_TRANS, infoP);
}

void impTPCCPayment(h,dataP,trpcStatus)
 encina_handle_t h;
 payment_data_t *dataP;
 trpc_status_t  *trpcStatus;
{
  static int numCalls = 0;
  thread_info_t *infoP = get_thread_data();
  FUNCTION_BEGIN("Payment", dataP, PAYMENT_TRANS, infoP);
  if (server_null_test) {
                sim_payment(dataP);
  } else {
                do_payment(infoP->cnP, dataP);
  }

  if (dataP->header.returncode != TPCC_SUCCESS) {
    logprintf("< impTPCCPayment; rc=%d, sql=%d, isam=%d\n",
                                dataP->header.returncode,
                                dataP->header.sql_code,
                                dataP->header.isam_code);
  }
  if (++numCalls % 10000 == 0) {
     err_printf("impTPCCPayment so far %d\n", numCalls);
  }
  FUNCTION_END("Payment", dataP, PAYMENT_TRANS, infoP);
}

void impTPCCOrderStatus(h,dataP,trpcStatus)
 encina_handle_t h;
 orderStatus_data_t *dataP;
 trpc_status_t  *trpcStatus;
{
  thread_info_t *infoP = get_thread_data();
  FUNCTION_BEGIN("OrderStatus", dataP, ORDER_STAT_TRANS, infoP);
  if (server_null_test) {
                sim_order_status(dataP);
  } else {
                do_order_status(infoP->cnP, dataP);
  }

  if (dataP->header.returncode != TPCC_SUCCESS) {
    logprintf("< impTPCCOrderStatus; rc=%d, sql=%d, isam=%d\n",
                                dataP->header.returncode,
                                dataP->header.sql_code,
                                dataP->header.isam_code);
  }
  FUNCTION_END("OrderStatus", dataP, ORDER_STAT_TRANS, infoP);
}


void impTPCCStockLevel(h,dataP,trpcStatus)
 encina_handle_t h;
 stockLevel_data_t *dataP;
 trpc_status_t   *trpcStatus;
{
  thread_info_t *infoP = get_thread_data();
  FUNCTION_BEGIN("StockLevel", dataP, STOCK_TRANS, infoP);
  if (server_null_test) {
                sim_stock_level(dataP);
```

```
  } else {
                do_stock_level(infoP->cnP, dataP);
  }

  if (dataP->header.returncode != TPCC_SUCCESS) {
    logprintf("< impTPCCStockLevel; rc=%d, sql=%d, isam=%d\n",
                                dataP->header.returncode,
                                dataP->header.sql_code,
                                dataP->header.isam_code);
  }
  FUNCTION_END("StockLevel", dataP, STOCK_TRANS, infoP);
}

void impTPCCDelivery(dataP,trpcStatus)
 delivery_data_t *dataP;
 trpc_status_t   *trpcStatus;
{
#ifdef WIN32
   delivery_sql_t dbData;
#endif

   thread_info_t *infoP = NULL;
   FUNCTION_BEGIN("DELIVERY", dataP, DELIVERY_TRANS, infoP);
   if (num_deferred_dvry_threads > 0) {
      queue_delivery(dataP);
   } else {
#ifdef WIN32
                if (server_null_test) {
                   sim_delivery(&dbData);
                } else {
                   infoP = get_thread_data();
                   do_delivery(infoP->cnP,&dbData);
                }
#else
                if (server_null_test) {
                   sim_delivery(dataP);
                } else {
                   infoP = get_thread_data();
                   do_delivery(infoP->cnP,dataP);
                }
#endif
   }


   if (dataP->header.returncode != TPCC_SUCCESS) {
     logprintf("< impTPCCDelivery;rc=%d, sql=%d, isam=%d\n",
                                dataP->header.returncode,
                                dataP->header.sql_code,
                                dataP->header.isam_code);
   }
   FUNCTION_END("DELIVERY", dataP, DELIVERY_TRANS, infoP);
}

/* functions in order to run with NULL database */
static void *create_null_connection() {
   static cn_num = 0;
   int *id = (int *)malloc(sizeof(int));
   *id = cn_num++;
   return id;
}

static void clean_null_connection(void*ptr) {
   free(ptr);
   return;
}
```

stocklevel.tacf

```
/*
 * Copyright (C) 1991, 1990 Transarc Corporation
 * All Rights Reserved
 */
/*
 * stocklevel.tacf -- attribute configuration file for tpcc server.
 * used for transparent binding
 *
 * $Revision: 1.1 $
 * $Date: 1998/11/06 21:10:16 $
 * $Log:          tpcc.tacf,v $
 *
 * $TALog: stocklevel.tacf,v $
 * Revision 1.1  1998/11/06  21:10:16  dongfeng
 * - Move all files common to client and server to tpcc/common
 *   directory
 * [added by delta dongfeng-23677-TPCC-new-directory-structures, r1.1]
 *
 * Revision 1.2  1998/10/08  18:03:04  gerstl
 * Changes to allow configurations where some servers only service
 * specific transaction types. Split transaction interfaces by type.
 * [added by delta gerstl-23515-TPCC-allow-separate-online-transaction-interfaces,r1.1]
 *
 *
 *
 */

[explicit_handle encina_handle_t]
interface stocklevel
{
}
```

stocklevel.tidl

```
/*
* id: $id: $
*
* component_name: encina benchmarks
*
* the following functions list may not be complete.
* functions defined by/via macros may not be included.
*
* functions:
*    <fill_me_in>
*
* origins: transarc corp.
*
* (c) copyright transarc corp. 1995, 1993
* all rights reserved
* licensed materials - property of transarc
*
* us government users restricted rights - use, duplication or
* disclosure restricted by gsa adp schedule contract with transarc corp
*/
/*
* history
* $talog: $
*/
/*
* stocklevel.tidl -- interface definition file for tpccserver.
*
* $revision: 1.0 $
* $date: 1995/10/20 21:55:05 $
* $log:          tpcc.tidl,v $
*/
[
  uuid(1dda58c8-5e05-11d2-bd18-9e621208aa77),
  version(1.0)
]
interface stocklevel
{
import "tpm/mon/mon_handle.idl";
import "tpcc_type.idl";

[nontransactional] void          impTPCCStockLevel(
                                        [in,out] stockLevel_data_t *dataP,
                                        [out] trpc_status_t  * trpcStatus);
}
```

tpcc.h

```
#if !defined(TPCC_H_INCLUDED)
#define TPCC_H_INCLUDED
/*****************************************************************************/
/*                                                                           */
/*      File: tpcc.h                                                         */
/*      created: 8-26-91                                                     */
/*                                                                           */
/*      program description:                                                 */
/*                                                                           */
/*      This module contains global variables and data definitions           */
/*      for the tpcc application.                                            */
/*                                                                           */
/*****************************************************************************/

#include "../../common/tpcc_type.h"

#define TPCCH

/*-------------------------------------------------------*/
/*            Global numbers, constants,...              */
/*-------------------------------------------------------*/

#define INVALID_ITEM           100
#define TRAN_OK                0
#define REMOTE_WAREHOUSE       17

#define FORM_DATE              1
#define FORM_DATETIME          2

#define MAX_ITEMS 15

/*-------------------------------------------------------*/
/*            transaction structures                     */
/*-------------------------------------------------------*/

typedef orderStatus_data_t  OrderStatus_data;
typedef newOrder_data_t NewOrder_data;
typedef stockLevel_data_t StockLevel_data;
```

```
typedef delivery_data_t Delivery_data;
typedef payment_data_t Payment_data;

/*****************************************************************************
 Compatibility for older.sqc files
 *****************************************************************************/
#define s_C_BALANCE c_balance
#define s_C_CITY c_city
#define s_C_CREDIT c_credit
#define s_C_CREDIT_LIM c_credit_lim
#define s_C_DATA c_data
#define s_C_DISCOUNT c_discount
#define s_C_D_ID c_d_id
#define s_C_FIRST c_first
#define s_C_ID c_id
#define s_C_LAST c_last
#define s_C_MIDDLE c_middle
#define s_C_PHONE c_phone
#define s_C_SINCE c_date
#define s_C_STATE c_state
#define s_C_STREET_1 c_street_1
#define s_C_STREET_2 c_street_2
#define s_C_W_ID c_w_id
#define s_C_ZIP c_zip
#define s_D_CITY d_city
#define s_D_ID d_id
#define s_D_STATE d_state
#define s_D_STREET_1 d_street_1
#define s_D_STREET_2 d_street_2
#define s_D_TAX d_tax
#define s_D_ZIP d_zip
#define s_H_AMOUNT h_amount
#define s_H_DATE pay_date
#define s_I_NAME name_i
#define s_I_PRICE price
#define s_OL_AMOUNT ol_amount
#define s_OL_DELIVERY_D delivery_date
#define s_OL_I_ID ol_i_id
#define s_OL_QUANTITY ol_quantity
#define s_OL_SUPPLY_W_ID ol_supply_w_id
#define s_O_CARRIER_ID o_carrier_id
#define s_O_ENTRY_D entry_date
#define s_O_ID o_id
#define s_O_OL_CNT o_ol_cnt
#define s_S_QUANTITY s_quantity
#define S_QUANTITY quantity
#define s_W_CITY w_city
#define s_W_ID w_id
#define s_W_STATE w_state
#define s_W_STREET_1 w_street_1
#define s_W_STREET_2 w_street_2
#define s_W_TAX w_tax
#define s_W_ZIP w_zip
#define s_all_local o_all_local
#define s_brand_generic brand_generic
#define s_exec_status exec_status
#define s_low_stock stock_count
#define s_ol_cnt o_ol_cnt
#define s_queued_time queued_time
#define s_status_line statusline
#define s_threshold threshold
#define s_total_amount total
#define s_transtatus header.returncode

#if 0
#define NEWORDER_SERVICE    "NEWORD"
#define PAYMENT_SERVICE     "PAYMENT"
#define DELIVERY_SERVICE    "DELIVERY"
#define STOCKLEVEL_SERVICE  "STOCKLEV"
#define ORDERSTATUS_SERVICE "ORDSTAT"
#else
#define NEWORDER_SERVICE    "neword_sql"
#define PAYMENT_SERVICE     "payment_sql"
#define DELIVERY_SERVICE    "delivery_sql"
#define STOCKLEVEL_SERVICE  "stocklev_sql"
#define ORDERSTATUS_SERVICE "ordstat_sql"
#endif

#endif /* TPCC_H_INCLUDED */
```

tpcc.tacf

```
/*
 * Copyright (C) 1991, 1990 Transarc Corporation
 * All Rights Reserved
 */
/*
 *  tpcc.tacf -- attribute configuration file for tpcc server.
 * used for transparent binding
 *
 * $Revision: 1.1 $
 * $Date: 1998/11/06 21:10:17 $
 * $Log:          tpcc.tacf,v $
Revision 4.2  95/05/16  10:55:49  10:55:49  tpcc (TPCC Benchmark)
Added necessary RCS ident strings

 */

[implicit_handle(mon_handle_t handle)]
interface tpccTransactions
{
}
```

tpcc_trans.tacf

```
/*
 * Copyright (C) 1991, 1990 Transarc Corporation
 * All Rights Reserved
 */
/*
 *  neworder.tacf -- attribute configuration file for tpcc server.
 * used for transparent binding
 *
 * $Revision: 1.1 $
 * $Date: 1998/11/06 21:10:17 $
 * $Log:          tpcc.tacf,v $
 *
 * $TALog: tpcc_trans.tacf,v $
 * Revision 1.1  1998/11/06  21:10:17  dongfeng
 * - Move all files common to client and server to tpcc/common
 *   directory
 * [added by delta dongfeng-23677-TPCC-new-directory-structures, r1.1]
 *
 * Revision 1.1  1997/06/16  22:04:48  oz
 * - Integration with Data Dependent Routing: Phase 1
 *   Separated the all the binding related code into its own files.
 * - Added mon_client_utils.[ch] that export binding related calls.
 * - Added a TPCC_USE_DDR compile time switch
 * - Added tpcc_trans.tidl: All the functions in one interface.
 * [added by delta oz-20170-TPCC-add-data-dependent-routing, r1.1]
 *
 *
 */

[implicit_handle(mon_handle_t handle)]
interface tpccTrans
{
}
```

tpcc_trans.tidl

```
/*
 * id: $id: $
 *
 * component_name: encina benchmarks
 *
 * the following functions list may not be complete.
 * functions defined by/via macros may not be included.
 *
 * functions:
 *    <fill_me_in>
 *
 * origins: transarc corp.
 *
 * (c) copyright transarc corp. 1995, 1993
 * all rights reserved
 * licensed materials - property of transarc
 *
 * us government users restricted rights - use, duplication or
 * disclosure restricted by gsa adp schedule contract with transarc corp
 */
/*
 * history
 * $talog: $
 */

/*
 * tpcc_trans.tidl -- interface definition file for tpccserver.
 *
 * $revision: 1.11 $
 * $date: 1995/10/20 21:55:05 $
 * $log:          tpcc.tidl,v $
 */

[uuid(955d7288-e672-11d0-bcef-9e621234aa77), version(1.0)]

interface tpccTrans
{
import "tpm/mon/mon_handle.idl";
import "tpcc_type.idl";

[nontransactional] void
                impTPCCNewOrder([in,out] newOrder_data_t *dataP,
                                                [out] trpc_status_t  * trpcStatus);

[nontransactional] void
                impTPCCPayment([in,out] payment_data_t *dataP,
                                                [out] trpc_status_t  * trpcStatus);

[nontransactional] void
                impTPCCOrderStatus([in,out] orderStatus_data_t  *dataP,
                                                [out] trpc_status_t  * trpcStatus);
```

```
[nontransactional] void
                impTPCCStockLevel([in,out] stockLevel_data_t *dataP,
                                                [out] trpc_status_t  * trpcStatus);


[nontransactional] void
        impTPCCNOInfo([out] dbInfo_data_t *dataP,
                [out]  trpc_status_t  * trpcStatus);

}
```

tpcc_type.idl

```
/*
 *                tpcc_type.idl
 *
 * $Revision: 1.2 $
 * $Date: 1998/12/08 18:55:21 $
 * $Log:            $
 *
 *
 * $TALog: tpcc_type.idl,v $
 * Revision 1.2  1998/12/08  18:55:21  wenjian
 * Add "int stats" to data_header structure
 * [from r1.1 by delta wenjian-23785-TPCC-pass-statsFrequency-from-client-to-server,r1.1]
 *
 * 2001/11/25 klavs: Changed warehouses references to long int
 *
 * Revision 1.1  1998/11/06  21:10:17  dongfeng
 * - Move all files common to client and server totpcc/common
 *   directory
 * [added by delta dongfeng-23677-TPCC-new-directory-structures, r1.1]
 *
 * Revision 1.11  1998/01/24  14:17:07  oz
 * - User server name to identify server and name delivery file
 * - Use env variable HOME instead of /home/encina if HOME is set
 *
 * - Added const ONLINE_INTERFACES
 * [from r1.10 by delta oz-21687-TPCC-use-server-name-to-identify-process, r1.1]
 *
 * Revision 1.10  1998/01/23  15:09:11  oz
 * - Updated the SP TPCC directory to the latest files used
 *   during the SP tpcc audit.
 * [from r1.9 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27,  r1.1]
 *
 *
 *
 */
[
uuid(008c6338-2b0a-1001-a9ab-02608c2f015a), version(1)
]
interface tpcc_types {

const long NAME_LENGTH = 32;

const long NEWO_INTERFACE =       0x01;
const long PAYMENT_INTERFACE =    0x02;
const long ORDER_STAT_INTERFACE = 0x04;
const long DELIVERY_INTERFACE =   0x08;
const long STOCK_INTERFACE =      0x10;
const long ONLINE_INTERFACES = NEWO_INTERFACE | PAYMENT_INTERFACE |
ORDER_STAT_INTERFACE | STOCK_INTERFACE;
const long ALL_INTERFACE =       0xffff;

const long NEWO_TRANS = 1;
const long PAYMENT_TRANS = 2;
const long ORDER_STAT_TRANS = 3;
const long DELIVERY_TRANS = 4;
const long STOCK_TRANS = 5;
const long MAX_TRAN_TYPE = 5;

typedef struct {
                long int sec;
                long int usec;
} time_type;

typedef struct  {
        short  int  dtype;
        short  int  returncode;
                long int   sql_code;
                long int   isam_code;
        long  int   num_rms;

        short  int  stats;       /*  For instrument only */
                time_type  start_time;          /*  For Debug Purposes only  */
                time_type  end_time;            /*  For Debug Purposes only  */
} data_header;

/* Definitions for payment transaction
 *
 *  payment_data_t
 *
 *  An in-out structure for payment transaction.
 *  It contains all the input parameters as well as the output parameters.
 */
typedef struct {
        data_header header;
        long  int w_id;
        short  int d_id;
        short  int c_id;
        long  int c_w_id;
        short  int c_d_id;
        short  int byname;
        double    h_amount;
        char      pay_date[20];

        char      w_name[11];
        char      w_street_1[21];
        char      w_street_2[21];
        char      w_city[21];
        char      w_state[3];
        char      w_zip[10];

        char      d_name[11];
        char      d_street_1[21];
        char      d_street_2[21];
        char      d_city[21];
        char      d_state[3];
        char      d_zip[10];

        char      c_first[17]; /* was C_LAST_LEN already includes +1*/
        char      c_middle[3];
        char      c_last[17];
        char      c_phone[17];
        char      c_credit[3];
        char      c_street_1[21];
        char      c_street_2[21];
        char      c_city[21];
        char      c_state[3];
        char      c_zip[10];
        double    c_credit_lim;
        double    c_balance;
        double    c_discount;
        double    c_ytd_payment;
        short  int c_payment_cnt;
        char      c_date[20];
        char      c_data[201];
} payment_data_t;

/* Definitions for new order transaction */

typedef struct {
        long  int  ol_supply_w_id;
        short  int  ol_quantity;
        short  int  s_quantity;
        long  int  ol_i_id;
        char       name_i[25];
        char       brand_generic[2];
        double     price;
        double     ol_amount;
                   long int   s_idx;
                   char       s_dist[25];
} OL_TABLE, newOrder_item_t;

typedef struct {
                   data_header     header;
                   long  int       w_id;
                   short int       d_id;
                   short int       c_id;
                   short int       o_ol_cnt;
                   short int       o_all_local;
                   short int       items_valid;/* true if all valid*/
                   short int          total_items;
                   long int        o_id;
        double          w_tax;
                   double          d_tax;
                   double          total;
                   double          c_discount;
                   char            entry_date[20];
                   char            c_last[17];
                   char            c_credit[3];
                   char               statusline[26];
                   OL_TABLE    item[15];
} newOrder_data_t;

/* Definitions for order status transaction */

typedef struct {
  long int  ol_i_id;
  long int  ol_supply_w_id;
  short int  ol_quantity;
  double     ol_amount;
  char       delivery_date[20];
} orderStatusItem_t;

typedef struct {
        data_header header;
        long  int w_id;
        short  int d_id;
        short  int c_id;
        short  int o_id;
        short  int o_ol_cnt;
        short  int byname;
        short      o_carrier_id;
        char       c_last[17];
```

```
        char      c_first[17];
        char      c_middle[3];
        char      entry_date[20];
        double    c_balance;
        orderStatusItem_t item[15];
} orderStatus_data_t;

/* Definitions for stock level transaction*/

typedef struct {
 data_header header;
 long  int w_id;
 short int d_id;
 short int threshold;
 long int  stock_count;
} stockLevel_data_t;

/* Definitions for delivery transaction*/

typedef struct {
 data_header header;
 long  int w_id;
 short int o_carrier_id;
 long int queued_time;
 short status;
 char exec_status[50];
 double start_queue;
} delivery_data_t;

typedef struct {
 long int first_wh;
 long int last_wh;
 long int server_id;
} dbInfo_data_t;

/*
 *  A union of all the transactions
 */
typedef union switch(long int tran_type) data {
 case NEWO_TRANS:           newOrder_data_t    new_order;
 case PAYMENT_TRANS:     payment_data_t     payment;
 case ORDER_STAT_TRANS:  orderStatus_data_t order_status;
 case DELIVERY_TRANS:    delivery_data_t    delivery;
 case STOCK_TRANS:          stockLevel_data_t  stock_level;
} tpcc_data_t;


}
```

tpcc_utils.c

```
/*
 *
 *      tpcc_utils.c
 *
 * $Revision: 1.2 $
 * $Date: 1998/12/14 20:27:57 $
 * $Log:        $
 *
 *
 *
 *
 * $TALog: tpcc_utils.c,v $
 * Revision 1.2  1998/12/14  20:27:57  wenjian
 * Made corresponding changes due to data structure change of tran_info_t.
 *
 * - Add header file winsock.h for NT platform
 * [from r1.1 by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.3]
 *
 * Revision 1.1  1998/12/11  16:37:58  wenjian
 * Move some common functions from client/client_utils.c to common/tpcc_utils.c.
 * In this version, we only move time_diff_ms().Need some work in order to
 * move other functions like ERROUT.
 *
 * - A file including utility functions for both client and server
 * [added by delta wenjian-23788-TPCC-use-single-stats-var-for-each-client-and-server, r1.2]
 *
 *
 *
 *    tpcc_utils.c
 *      Generic utilities used by the client and server processes
 */

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

#if defined (solaris)
#include <dce/pthread.h>
#else   /* solaris */
#include <pthread.h>
#endif

#include "databuf.h"
#include "do_tpcc.h"
#include "tpcc_type.h"

#ifdef WIN32
#include <winsock.h>
#endif

/*
 *  time_diff_ms
 *      Return the difference in miliseconds between two times
 */
int time_diff_ms(t2,t1)
  struct timeval *t2, *t1;
{
    int t_diff;

    t_diff = (t2->tv_usec + 1000000 - t1->tv_usec + 500) / 1000 +
            (t2->tv_sec - t1->tv_sec - 1) * 1000;

    return(t_diff);
}
```

util.h

```
#ifndef LOCAL_UTIL_H
#define LOCAL_UTIL_H

#include "util_token.h"

#define UTIL_ALLOC(ptr, type, size)                     \
    ptr = (type)malloc(size);                                                   \
    if (ptr==NULL) {                                                                    \
                fprintf(stderr, "UTIL_ALLOC failed\n");                 \
                exit(1);                                                                \
    }

#endif
```

util_alloc.h

```
/*
 * util_alloc.h
 *
 * $Revision: 1.1 $
 * $Date: 1998/11/06 21:10:18 $
 * $Log:        util_alloc.h,v $
 * Revision 4.2  95/05/16  10:55:43  10:55:43  tpcc (TPCC Benchmark)
 * Added necessary RCS ident strings
 *
 *
 */

#ifndef TRANSARC_UTIL_ALLOC_H
#define TRANSARC_UTIL_ALLOC_H

/*
 * UTIL_[ALLOC, REALLOC, NEW, FREE] -- macros that wrap calls to
 * malloc, realloc, free. The allocation macros check the return
 * value, a NULL pointer is converted into a fatal error.
 */
#define UTIL_ALLOC_ROBUST(ptr, type, size)                                  \
    ((ptr) = (type) malloc(size))

#define UTIL_ALLOC(ptr, type, size)                                         \
do {
\
    if (UTIL_ALLOC_ROBUST(ptr, type, size) == 0)                    \
                util_MemoryError("UTIL_ALLOC", __FILE__, __LINE__);             \
} while (0)

#define UTIL_REALLOC_ROBUST(ptr, type, size)                                \
    (ptr = (type) realloc((void *) ptr, size))

#define UTIL_REALLOC(ptr, type, size)                                       \
do {
\
    if (UTIL_REALLOC_ROBUST(ptr, type, size) == 0)                  \
                util_MemoryError("UTIL_REALLOC", __FILE__, __LINE__);           \
} while (0)

#define UTIL_FREE(ptr)                                                      \
do {
\
    if (!ptr) {                                                                 \
                util_MemoryError("UTIL_FREE", __FILE__, __LINE__); \
    }
\
    free((void *) (ptr));                                                       \
    ptr = 0;        /* Make all free'd pointers zero. */                \
} while (0)

#define         UTIL_ALLOC_ARRAY_ROBUST(ptr, type, number)                  \
    ((ptr) = (type *)malloc(sizeof(type) * (number)))

#define         UTIL_ALLOC_ARRAY(ptr, type, number)                         \
do {
\
    if (UTIL_ALLOC_ARRAY_ROBUST(ptr, type, number) == 0)            \
```

```
                util_MemoryError("UTIL_ALLOC_ARRAY",__FILE__, __LINE__); \
} while (0)

#define         UTIL_COPY_STRING_ROBUST(to,from)                                \
  (((to) = (char *)malloc(strlen((char*)(from))+1)) ?       \
    strcpy((char *)(to), (char *)(from)) : 0)

#define         UTIL_COPY_STRING(to, from)                                                      \
do {
\
  if (UTIL_COPY_STRING_ROBUST(to, from) == 0)                                   \
                util_MemoryError("UTIL_COPY_STRING", __FILE__, __LINE__); \
} while (0)


#endif /* TRANSARC_UTIL_ALLOC_H */
```

util_token.h

```
/*
* ID: $Id: util_token.h,v 1.1 1998/11/06 21:10:18 dongfeng Exp $
*
* COMPONENT_NAME: Encina Toolkit Executive
*
* The following functions list may not be complete.
* Functions defined by/via macros may not be included.
*
* FUNCTIONS:
*
* ORIGINS: Transarc Corp.
*
* (C) COPYRIGHT Transarc Corp. 1995, 1994, 1993, 1990
* All Rights Reserved
* Licensed Materials - Property of Transarc
*
* US Government Users Restricted Rights - Use, duplication or
* disclosure restricted by GSA ADP Schedule Contract with Transarc Corp
*/
/*
* HISTORY
* $TALog: util_token.h,v $
* Revision 1.1  1998/11/06  21:10:18  dongfeng
* - Move all files common to client and server totpcc/common
*   directory
* [added by delta dongfeng-23677-TPCC-new-directory-structures, r1.1]
*
* Revision 1.7  1995/01/30  12:50:13  barry
* Update copyrights for Encina 1.2.
* [from r1.6 by delta barry-0000-update-copyrights-for-1.2, r1.1]
*
* Revision 1.6  1995/01/13  14:12:51  psu
* fix comment to conform to coding standards.
* [from r1.5 by delta psu-13196-client-interoperate-with-oracle-pro-c-2, r1.3]
*
* Revision 1.5  1995/01/12  20:34:22  psu
* put comment on ## fix in the code
*
* try to make sure people don't change the use of ##
* [from r1.4 by delta psu-13196-client-interoperate-with-oracle-pro-c-2, r1.2]
*
* Revision 1.4  1995/01/12  15:48:28  psu
* fix use of ## to make pro*c happy
*
* pro*c 2.0 can't deal with a ## b ## c, change to a ## b##c.
* [from r1.3 by delta psu-13196-client-interoperate-with-oracle-pro-c-2, r1.1]
*
* Revision 1.3  1994/02/04  17:22:22  pinaki
* Update copyright.
* [from r1.2 by delta pinaki-0000-update-copyright-for-1.1, r1.1]
*
* Revision 1.2  1993/12/18  22:06:57  mwyoung
* [from r1.1 by delta mwyoung-10043-util-always-offer-UTIL_IDENT,r1.1]
*
* Revision 1.1  1993/12/03  22:00:04  mwyoung
* Split the various features into separate files, so that they can
* be included separately.
* [added by delta mwyoung-9848-utils-split-util.h-into-separately-usable-parts,r1.1]
*
*/
#ifndef TRANSARC_UTIL_TOKEN_H
#define TRANSARC_UTIL_TOKEN_H

#include <encina/c_prologue.h>

/* UTIL_IDENT -- the identity function */
#define UTIL_IDENT(a)                           a

/*
* UTIL_[STRING, CONCAT, CONCAT3] -- macros for converting into, and
* concatenating together, strings.
*/

/* Note, the a ## b##c is needed to make some broken cpp's work correctly.
* This was originally put here for OraclePro*C 2.0, but other compilers
* may have similar problems.
*/
#if                     ENCINA_C_ANSI_STRING_TOKEN_SUPPORT
#define UTIL_STRING(a)     # a
#define UTIL_CONCAT(a, b)                       a ## b
#define         UTIL_CONCAT3(a,b,c)     a ## b##c
#else           /* ENCINA_C_ANSI_STRING_TOKEN_SUPPORT */
#define UTIL_STRING(a)                          "a"
#define UTIL_CONCAT(a, b)                       UTIL_IDENT(a)b
#define         UTIL_CONCAT3(a,b,c)     UTIL_CONCAT(a,b)c
#endif          /* ENCINA_C_ANSI_STRING_TOKEN_SUPPORT */

#include <encina/c_epilogue.h>
#endif /* TRANSARC_UTIL_TOKEN_H */
```

utilities.h

```
/*
 * ID: $Id: utilities.h,v 1.1 1998/11/06 21:10:19 dongfeng Exp $
 *
 * COMPONENT_NAME: Encina Toolkit Server Core
 *
 * ORIGINS: Transarc Corp.
 *
 * (C) COPYRIGHT Transarc Corp. 1995
 * All Rights Reserved
 * Licensed Materials - Property of Transarc
 *
 * US Government Users Restricted Rights - Use, duplication or
 * disclosure restricted by GSA ADP Schedule Contract with Transarc Corp
 * $Revision: 1.1 $
 * $Log:        utilities.h,v $
 *
 *
 * $TALog: utilities.h,v $
 * Revision 1.1  1998/11/06  21:10:19  dongfeng
 * - Move all files common to client and server totpcc/common
 *   directory
 * [added by delta dongfeng-23677-TPCC-new-directory-structures, r1.1]
 *
 * Revision 1.7  1998/10/27  14:57:52  dongfeng
 * Change enc_status to a data structure that has fields:
 *   - Status code
 *   - Line Number
 *   - File Name
 *   - Encina Error Code
 *   - Error Msg
 * Remove statusMsgs in web_tpcc.c
 * [from r1.6 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.6]
 *
 * Revision 1.6  1998/10/22  19:18:37  dongfeng
 * [from r1.5 by delta dongfeng-23067-TPCC-add-web-based-tpcc-client, r1.2]
 *
 * Revision 1.5  1998/01/23  15:09:16  oz
 * - Updated the SP TPCC directory to the latest files used
 *   during the SP tpcc audit.
 * [from r1.4 by delta oz-20774-TPCC-update-to-latest-SP-version-11-27, r1.1]
 *
 *
 */

/*
 * utilities.h -- holds declarations, macros, and constants used by the
 * telshop/merchandise client-server program.
 *
 * $Date: 1998/11/06 21:10:19 $
 */

#ifndef _UTILITIES_H_
#define _UTILITIES_H_

#include <dce/rpc.h>
#include <dce/dce_error.h>
#include <encina/encina.h>
#include <stdlib.h>

#include <utils/trace.h>
#include "util_alloc.h"

/*  Boolean type, and its constants  */

#define FALSE 0
#define TRUE  1

#if       ENCINA_C_ANSI_STRING_TOKEN_SUPPORT
#define UTIL_STRING(a)          # a
#define UTIL_CONCAT(a, b)       a ## b
#define UTIL_CONCAT3(a,b,c)     a ## b##c
#else   /* ENCINA_C_ANSI_STRING_TOKEN_SUPPORT */
#define UTIL_STRING(a)          "a"
#define UTIL_CONCAT(a, b)       UTIL_IDENT(a)b
#define UTIL_CONCAT3(a,b,c)     UTIL_CONCAT(a,b)c
#endif  /* ENCINA_C_ANSI_STRING_TOKEN_SUPPORT */

/*  ENCINA_CALL:  Make fail-fast calls on the various services. */

/*  Macro delimiters  */
```

```
#define BEGIN_MACRO do {

#define END_MACRO } while (0)

/*  FATAL --  Failure.  Print error message and exit the program  */

void exit_program();

#ifndef FATAL
#define FATAL(args)
\
BEGIN_MACRO
                \
   printf args;                                                                 \
   exit(1);
\
END_MACRO
#endif

/*  ENCINA_CALL:  Make fail-fast calls on the various services. */

#define ENCINA_CALL_RC(proc_name,call,rc)                       \
BEGIN_MACRO                                             \
   char _errorMsg[ENCINA_MAX_STATUS_STRING_SIZE];                               \
   rc = (call);                                                                 \
   if (rc)  {                                                   \
      encina_StatusToString(rc, ENCINA_MAX_STATUS_STRING_SIZE,          \
                                                _errorMsg);
\
      err_printf( "%x \n", rc);                                                 \
      err_printf( "%s \n", _errorMsg);                                          \
      err_printf( "%s \n", proc_name);                                          \
   }
                \
END_MACRO

#define ENCINA_CALL(proc_name,call)                     \
BEGIN_MACRO                                                             \
   unsigned long _status;                                       \
   ENCINA_CALL_RC(proc_name,call,_status);                      \
   if (_status) exit_program(_status);                          \
END_MACRO

typedef enum {
   action_exit,
   action_continue
} error_action_t;

#define CHECK_DCE_STATUS(_status, _msg, _action)                \
{                                                       \
   int error_stat;                                      \
   unsigned long _rc = (_status);                       \
   unsigned char error_string[dce_c_error_string_len];          \
   if ((_status) != rpc_s_ok) {                         \
                dce_error_inq_text(_rc, error_string, &error_stat);         \
                err_printf("%s failed, error: %s (%d)\n", _msg, error_string, _rc);\
                if ((_action) == action_exit)                   \
                   exit(-1);                                    \
   }                                                    \
}

#define DCE_CALL(call, args)                            \
{                                                       \
   call args;                                           \
   CHECK_DCE_STATUS(status, UTIL_STRING(call), action_exit);\
}



/*  MALLOC_CHECK --  Make sure there is memory to be allocated;
 *  fail if there is not.  */

#define MALLOC_CHECK(memP)                                              \
BEGIN_MACRO                                                             \
   if (!(memP))                                                         \
      FATAL(("Out of memory.\n"));                              \
END_MACRO

/*  ASSERT --  internal checks that assure the program is running correctly.
 *  Use to check program correctness, not user input.  */

#ifndef ASSERT
#define ASSERT(condition)                                               \
BEGIN_MACRO
                \
   if (!(condition))
\
      FATAL(("%s (%d):  Assertion failed.\n", __FILE__, __LINE__));     \
END_MACRO
#endif

#define RAND(lim1, lim2)  ((int)(drand48()*((lim2)-(lim1)+1))+ (lim1))

#ifndef BAD_STATUS
#define BAD_STATUS(call, status)                        \
   BEGIN_MACRO                                          \
      char _errorMsg[ENCINA_MAX_STATUS_STRING_SIZE];            \
      encina_StatusToString(status, ENCINA_MAX_STATUS_STRING_SIZE,      \
                  _errorMsg);                           \
      logprintf("%s: %s (%d)\n", UTIL_STRING(call), _errorMsg, status);   \
      exit(1);                                                  \
   END_MACRO
#endif

#ifndef boolean_t
#define boolean_t int
#endif

#ifndef EXPORT
#define EXPORT
#endif

#ifndef IMPORT
#define IMPORT extern
#endif

/* For web_tpcc_client */
#define CHK_STATUS(st, val, _errMsg)                    \
BEGIN_MACRO                                             \
    if(st) {                                            \
       enc_status.status=val;                           \
       strcpy(enc_status.file, __FILE__);               \
       enc_status.line= __LINE__;                       \
       enc_status.encinaError = st;                     \
       if(_errMsg)strcpy(enc_status.errorMsg, _errMsg);         \
       if(st!=1) return;  \
    }                                                   \
END_MACRO

#endif /*  _UTILITIES_H_  */
```

A.2 Client Transaction Code

initpay.sql

```
CREATE OR REPLACE PACKAGE initpay
AS
 TYPE rowidarray IS TABLE OF ROWID INDEX BY BINARY_INTEGER;
 row_id                rowidarray;
 cust_rowid             ROWID;
 dist_name             VARCHAR2(11);
 ware_name              VARCHAR2(11);
 c_num                 BINARY_INTEGER;
 PROCEDURE pay_init;
END initpay;
/
CREATE OR REPLACE PACKAGE BODY initpay AS
  PROCEDURE pay_init IS
  BEGIN
    NULL;
  END pay_init;
END initpay;
/
exit
```

payz.sql

```
DECLARE /* payz */
    not_serializable        EXCEPTION;
    PRAGMA EXCEPTION_INIT(not_serializable,-8177);
    deadlock                EXCEPTION;
    PRAGMA EXCEPTION_INIT(deadlock,-60);
    snapshot_too_old        EXCEPTION;
    PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);
  BEGIN
    LOOP BEGIN
      UPDATE ware
        SET w_ytd = w_ytd+:h_amount
        WHERE w_id = :w_id
        RETURNING w_name,
                w_street_1, w_street_2, w_city, w_state, w_zip
          INTO inittpcc.ware_name,
              :w_street_1, :w_street_2, :w_city, :w_state, :w_zip;


      SELECT rowid
      BULK COLLECT INTO inittpcc.row_id
      FROM cust
      WHERE c_d_id = :c_d_id AND c_w_id = :c_w_id AND c_last = :c_last
      ORDER BY c_last, c_d_id, c_w_id, c_first;

     inittpcc.c_num := sql%rowcount;
     inittpcc.cust_rowid := inittpcc.row_id((inittpcc.c_num)/ 2);
```

```
    UPDATE cust
     SET c_balance = c_balance - :h_amount,
         c_ytd_payment = c_ytd_payment+ :h_amount,
         c_payment_cnt = c_payment_cnt+1
    WHERE rowid = inittpcc.cust_rowid
    RETURNING
          c_id, c_first, c_middle, c_last, c_street_1, c_street_2,
          c_city, c_state, c_zip, c_phone,
          c_since, c_credit, c_credit_lim,
          c_discount, c_balance
       INTO :c_id, :c_first, :c_middle, :c_last,
            :c_street_1, :c_street_2, :c_city, :c_state,
            :c_zip, :c_phone, :c_since, :c_credit,
            :c_credit_lim, :c_discount, :c_balance;

    :c_data := ' ';
    IF :c_credit = 'BC' THEN
      UPDATE cust
        SET c_data = substr ((to_char (:c_id) || ' ' ||
                          to_char (:c_d_id) || ' ' ||
                          to_char (:c_w_id) || ' ' ||
                          to_char (:d_id) || ' ' ||
                          to_char (:w_id) || ' ' ||
                          to_char (:h_amount/100, '9999.99') || ' ' | ')
                         ||  c_data, 1, 500)
        WHERE rowid = inittpcc.cust_rowid
        RETURNING substr(c_data,1, 200)
        INTO :c_data;

    END IF;

    UPDATE dist
      SET d_ytd = d_ytd+:h_amount
      WHERE d_id = :d_id
       AND d_w_id = :w_id
      RETURNING  d_name, d_street_1, d_street_2, d_city,
                    d_state, d_zip
      INTO inittpcc.dist_name, :d_street_1, :d_street_2, :d_city,
          :d_state, :d_zip;

      IF  SQL%NOTFOUND
                THEN
                          raise NO_DATA_FOUND;
      END IF;

    INSERT INTO hist (h_c_id, h_c_d_id, h_c_w_id, h_d_id, h_w_id,
                    h_amount, h_date, h_data)
       VALUES (:c_id, :c_d_id, :c_w_id, :d_id, :w_id, :h_amount,
       :cr_date, inittpcc.ware_name|| '    ' || inittpcc.dist_name);

    EXIT;

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

  END LOOP;
END;
```

paynz.sql

```
DECLARE /* paynz */
    not_serializable       EXCEPTION;
    PRAGMA EXCEPTION_INIT(not_serializable,-8177);
    deadlock               EXCEPTION;
    PRAGMA EXCEPTION_INIT(deadlock,-60);
    snapshot_too_old       EXCEPTION;
    PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);
  BEGIN
    LOOP BEGIN
        UPDATE ware
          SET w_ytd = w_ytd + :h_amount
         WHERE w_id = :w_id
      RETURNING w_name, w_street_1, w_street_2, w_city, w_state, w_zip
          INTO inittpcc.ware_name, :w_street_1, :w_street_2, :w_city,
              :w_state, :w_zip;

        UPDATE  cust
          SET  c_balance = c_balance - :h_amount,
              c_ytd_payment = c_ytd_payment + :h_amount,
              c_payment_cnt = c_payment_cnt+1
      WHERE    c_id = :c_id AND c_d_id = :c_d_id AND
              c_w_id = :c_w_id
      RETURNING rowid, c_first, c_middle, c_last, c_street_1,
              c_street_2, c_city, c_state, c_zip, c_phone,
              c_since, c_credit, c_credit_lim,
              c_discount, c_balance
         INTO inittpcc.cust_rowid,:c_first, :c_middle, :c_last, :c_street_1,
              :c_street_2, :c_city, :c_state, :c_zip, :c_phone,
              :c_since, :c_credit, :c_credit_lim,
              :c_discount, :c_balance;
       IF SQL%NOTFOUND THEN
         raise NO_DATA_FOUND;
       END IF;


       IF :c_credit = 'BC' THEN
         UPDATE cust
                    SET c_data = substr ((to_char (:c_id) || ' ' ||
                        to_char (:c_d_id) || ' ' ||
                        to_char (:c_w_id) || ' ' ||
                        to_char (:d_id) || ' ' ||
                        to_char (:w_id) || ' ' ||
                        to_char (:h_amount/100, '9999.99') || ' ' | ')
                       ||  c_data, 1, 500)
          WHERE rowid = inittpcc.cust_rowid
       RETURNING substr(c_data,1, 200)
          INTO :c_data;

       END IF;

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


       INSERT INTO hist  (h_c_id, h_c_d_id, h_c_w_id, h_d_id, h_w_id,
                    h_amount, h_date, h_data)
       VALUES
         (:c_id, :c_d_id, :c_w_id, :d_id, :w_id, :h_amount,
          :cr_date, inittpcc.ware_name|| '    ' || inittpcc.dist_name);
       EXIT;

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

     END LOOP;
   END;
```

pldel.c

```
  #ifdef RCSID
static char *RCSid =
  "$Header: pldel.c 7030100.5 96/06/24 16:26:06 plai Generic<base> $ Copyr (c) 1994 Oracle";
#endif /* RCSID */

/*=================================================================+
|       Copyright (c) 1996  Oracle Corp, Redwood Shores, CA       |
|                  OPEN SYSTEMS PERFORMANCE GROUP                 |
|                     All Rights Reserved                         |
+=================================================================+
| FILENAME
|    pldel.c
| DESCRIPTION
|    OCI version of DELIVERY transaction in TPC-C benchmark.
+=================================================================*/

#include "tpcc.h"
#include "plora.h"
#ifdef TUX
#include <userlog.h>
#endif

/*
extern int userlog();
*/

#include "tpccflags.h"


#define DMLRETDEL

#define SQLTXT "BEGIN inittpcc.init_del; END;"

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

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

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

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

```
 c_d_id = :d_id AND c_id = :c_id"

#define NDISTS 10
#define ROWIDLEN 20


struct delctx {
 sb2 del_o_id_ind[NDISTS];
 sb2 d_id_ind[NDISTS];
 sb2 c_id_ind[NDISTS];
 sb2 del_date_ind[NDISTS];
 sb2 carrier_id_ind[NDISTS];
 sb2 amt_ind[NDISTS];

 ub4 del_o_id_len[NDISTS];
 ub4 c_id_len[NDISTS];
 int oid_ctx;
 int cid_ctx;
 OCIBind *olamt_bp;

 ub2 w_id_len[NDISTS];
 ub2 d_id_len[NDISTS];
 ub2 del_date_len[NDISTS];
 ub2 carrier_id_len[NDISTS];
 ub2 amt_len[NDISTS];

 ub2 del_o_id_rcode[NDISTS];
 ub2 cons_rcode[NDISTS];
 ub2 w_id_rcode[NDISTS];
 ub2 d_id_rcode[NDISTS];
 ub2 c_id_rcode[NDISTS];
 ub2 del_date_rcode[NDISTS];
 ub2 carrier_id_rcode[NDISTS];
 ub2 amt_rcode[NDISTS];

 int del_o_id[NDISTS];
 int del_d_id[NDISTS];
 int cons[NDISTS];
 int w_id[NDISTS];
 int d_id[NDISTS];
 int c_id[NDISTS];
 int carrier_id[NDISTS];
 int amt[NDISTS];
 ub4 del_o_id_rcnt;
 int retry;
 OCIRowid *no_rowid_ptr[NDISTS];
 OCIRowid *o_rowid_ptr[NDISTS];
 OCIDate del_date[NDISTS];
 OCIStmt *curd0;
 OCIStmt *curd1;
 OCIStmt *curd2;
 OCIStmt *curd3;
 OCIStmt *curd4;
 OCIStmt *curd5;
 OCIStmt *curd6;
 OCIStmt *curdtest;

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

 int norow;
};

typedef struct delctx delctx;
struct pldelctx {

 ub2 del_d_id_len[NDISTS];
 ub2 del_o_id_len[NDISTS];


 ub2 w_id_len;
 ub2 d_id_len[NDISTS];
 ub2 o_c_id_len[NDISTS];
 ub2 sums_len[NDISTS];
 ub2 carrier_id_len;
 ub2 ordcnt_len;
 ub2 del_date_len;

 int del_o_id[NDISTS];
 int del_d_id[NDISTS];
 int o_c_id[NDISTS];
 int sums[NDISTS];
 OCIDate del_date;
 int carrier_id;
 int ordcnt;

 ub4 del_o_id_rcnt;
 ub4 del_d_id_rcnt;
 ub4 o_c_id_rcnt;
 ub4 sums_rcnt;

 int retry;
 OCIStmt *curp1;
 OCIStmt *curp2;
 OCIBind *w_id_bp;
 OCIBind *d_id_bp;
 OCIBind *o_id_bp;
 OCIBind *o_c_id_bp;
 OCIBind *ordcnt_bp;
 OCIBind *sums_bp;
 OCIBind *del_date_bp;
 OCIBind *carrier_id_bp;
 OCIBind *retry_bp;

 int norow;

};
typedef struct pldelctx pldelctx;



#ifdef DMLRETDEL
struct amtctx {
 int ol_amt[NITEMS];
 sb2 ol_amt_ind[NITEMS];
 ub4 ol_amt_len[NITEMS];
 ub2 ol_amt_rcode[NITEMS];
 int ol_cnt;
};
typedef struct amtctx amtctx;

#endif


#ifdef DMLRETDEL
/*
sb4 no_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
            dvoid **bufpp, ub4 *alenp, ub1 *piecep,
            dvoid **indpp)
{
 *bufpp = (dvoid*)0;
 *alenp =0;
 *indpp = (dvoid*)0;
 *piecep =OCI_ONE_PIECE;
 return (OCI_CONTINUE);
}


sb4 TPC_oid_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
            dvoid **bufpp, ub4 **alenp, ub1 *piecep,
            dvoid **indpp, ub2 **rcodepp)
{
 *bufpp = &dctx->del_o_id[iter];
 *indpp= &dctx->del_o_id_ind[iter];
 dctx->del_o_id_len[iter]=sizeof(dctx->del_o_id[0]);
 *alenp= &dctx->del_o_id_len[iter];
 *rcodepp = &dctx->del_o_id_rcode[iter];
 *piecep =OCI_ONE_PIECE;
 return (OCI_CONTINUE);
}
sb4 cid_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
            dvoid **bufpp, ub4 **alenp, ub1 *piecep,
            dvoid **indpp, ub2 **rcodepp)
{
 *bufpp = &dctx->c_id[iter];
 *indpp= &dctx->c_id_ind[iter];
 dctx->c_id_len[iter]=sizeof(dctx->c_id[0]);
 *alenp= &dctx->c_id_len[iter];
 *rcodepp = &dctx->c_id_rcode[iter];
 *piecep =OCI_ONE_PIECE;
 return (OCI_CONTINUE);
}


# ifdef OLD
sb4 amt_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
            dvoid **bufpp, ub4 **alenp, ub1 *piecep,
            dvoid **indpp, ub2 **rcodepp)
{
```

```
amtctx *actx;
actx =(amtctx*)ctxp;
actx->ol_cnt=actx->ol_cnt+1;
*bufpp = &actx->ol_amt[index];
*indpp= &actx->ol_amt_ind[index];
actx->ol_amt_len[index]=sizeof(actx->ol_amt[0]);
*alenp= &actx->ol_amt_len[index];
*rcodepp = &actx->ol_amt_rcode[index];
*piecep =OCI_ONE_PIECE;
if (iter ==1 )
  return (OCI_CONTINUE);
else
  return (OCI_ERROR);
}
# else
sb4 amt_data(dvoid *ctxp, OCIBind *bp, ub4 iter, ub4 index,
             dvoid **bufpp, ub4 **alenp, ub1 *piecep,
             dvoid **indpp, ub2 **rcodepp)
{
amtctx *actx;
actx =(amtctx*)ctxp;
*bufpp = &actx->ol_amt[index];
*indpp= &actx->ol_amt_ind[index];
actx->ol_amt_len[index]=sizeof(actx->ol_amt[0]);
*alenp= &actx->ol_amt_len[index];
*rcodepp = &actx->ol_amt_rcode[index];
*piecep =OCI_ONE_PIECE;
return (OCI_CONTINUE);
}
# endif

*/
#endif


tkvcdinit (ora_cn_data_t *ora_SlotDataP)

{

 text stmbuf[SQL_BUF_SIZE];

 delctx *dctx;
 pldelctx *pldctx;
 amtctx *actx;
 global_delivery_t*delP;
 OCIEnv *tpcenv = ora_SlotDataP->tpcenv;
 OCIServer *tpcsrv = ora_SlotDataP->tpcsrv;
 OCIError *errhp = ora_SlotDataP->errhp;
 OCISvcCtx *tpcsvc = ora_SlotDataP->tpcsvc;
 OCISession *tpcusr = ora_SlotDataP->tpcusr;
 OCIStmt *curi = ora_SlotDataP->curi;

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

 ora_SlotDataP->dctx = (void *)dctx;
 delP = (global_delivery_t*)malloc(sizeof(global_delivery_t));
 memset(delP, (char)0, sizeof(global_delivery_t));
 ora_SlotDataP->delP = delP;

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


   DISCARD OCIHandleAlloc(tpcenv,(dvoid**)&pldctx->curp2, OCI_HTYPE_STMT,
                  0,  (dvoid**)0);
#if defined(ISO5) || defined(ISO6) || defined(ISO8)
 #if defined(ISO5)
    sqlfile("tkvcpdel_iso5.sql",stmbuf);
 #endif
 #if defined(ISO6)
    sqlfile("tkvcpdel_iso6.sql",stmbuf);
 #endif
 #if defined(ISO8)
    sqlfile("tkvcpdel_iso8.sql",stmbuf);
 #endif
#else
    sqlfile("tkvcpdel.sql",stmbuf);
#endif
   DISCARD OCIStmtPrepare(pldctx->curp2, errhp, stmbuf,
                  (ub4)strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT);
   OCIBNDPL(pldctx->curp2, pldctx->w_id_bp , errhp,":w_id",
            ADR(delP->w_id), SIZ(int), SQLT_INT,&pldctx->w_id_len);
   OCIBNDPL(pldctx->curp2, pldctx->ordcnt_bp , errhp,":ordcnt",
            ADR(pldctx->ordcnt), SIZ(int), SQLT_INT,&pldctx->ordcnt_len);

   OCIBNDPL(pldctx->curp2, pldctx->del_date_bp,errhp,":now",
            ADR(pldctx->del_date), SIZ(OCIDate),SQLT_ODT,&pldctx->del_date_len);

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

   OCIBNDPLA(pldctx->curp2, pldctx->d_id_bp, errhp,":d_id",
             pldctx->del_d_id, SIZ(int),SQLT_INT, pldctx->del_d_id_len,
             NDISTS, &pldctx->del_d_id_rcnt);
   OCIBNDPLA(pldctx->curp2, pldctx->o_id_bp, errhp,":order_id",
             pldctx->del_o_id,SIZ(int),SQLT_INT, pldctx->del_o_id_len,NDISTS,
             &pldctx->del_o_id_rcnt);
   OCIBNDPLA(pldctx->curp2, pldctx->sums_bp, errhp,":sums",
             pldctx->sums,SIZ(int),SQLT_INT, pldctx->sums_len,NDISTS,
             &pldctx->sums_rcnt);
   OCIBNDPLA(pldctx->curp2, pldctx->o_c_id_bp, errhp,":o_c_id",
             pldctx->o_c_id,SIZ(int),SQLT_INT, pldctx->o_c_id_len,NDISTS,
             &pldctx->o_c_id_rcnt);
   OCIBND(pldctx->curp2, pldctx->retry_bp , errhp,":retry",
          ADR(pldctx->retry), SIZ(int),SQLT_INT);

  return (0);

}



tkvcd (ora_cn_data_t *ora_SlotDataP)

{

  int i, j;
  int rpc,rcount,count;
  int invalid;

  delctx *dctx = (delctx *)ora_SlotDataP->dctx;
  pldelctx *pldctx = (pldelctx *)ora_SlotDataP->pldctx;
#ifdef DMLRETDEL /* VMM 1/13/98 */
  amtctx *actx = (amtctx *)ora_SlotDataP->actx;
#endif /* DMLRETDEL */
  global_delivery_t*delP = ora_SlotDataP->delP;
  OCIEnv *tpcenv = ora_SlotDataP->tpcenv;
  OCIServer *tpcsrv = ora_SlotDataP->tpcsrv;
  OCIError *errhp = ora_SlotDataP->errhp;
  OCISvcCtx *tpcsvc = ora_SlotDataP->tpcsvc;
  OCISession *tpcusr = ora_SlotDataP->tpcusr;
  OCIStmt *curi = ora_SlotDataP->curi;


  if (delP->plsqlflag)
  {

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

    pldctx->retry=0;

    DISCARD OCIERROR(errhp,
      OCIStmtExecute(tpcsvc,pldctx->curp2,errhp,1,0,NULLP(CONST OCISnapshot),
                NULLP(OCISnapshot),OCI_DEFAULT));
    for (i = 0; i < NDISTS; i++)
    {
      delP->del_o_id[i] = 0;
    }
    for (i = 0; i < pldctx->del_o_id_rcnt; i++)
      delP->del_o_id[pldctx->del_d_id[i]- 1] = pldctx->del_o_id[i];
  } else {
                return (-1);
  }
  return (0);

}

void tkvcddone (ora_cn_data_t *ora_SlotDataP)

{
  delctx *dctx = (delctx *)ora_SlotDataP->dctx;
  pldelctx *pldctx = (pldelctx *)ora_SlotDataP->pldctx;
  global_delivery_t*delP = ora_SlotDataP->delP;

  if (delP->plsqlflag)
  {
    if (pldctx)
    {
      DISCARD OCIHandleFree((dvoid *)pldctx->curp1,OCI_HTYPE_STMT);
      DISCARD OCIHandleFree((dvoid *)pldctx->curp2,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);
    ora_SlotDataP->delP = NULL;
  }
 if (delP) {
   free(delP);
   ora_SlotDataP->delP = NULL;
 }
}
}
```

plnew.c

```
 #ifdef RCSID
static char *RCSid =
 "$Header: tkvcnew.c 21-apr-98.18:32:59 rdecker Exp $ Copyr (c) 1994 Oracle";
#endif /* RCSID */

/*==============================================================+
|       Copyright (c) 1996 , 1997, 1998 Oracle Corp, Redwood Shores, CA       |
|                OPEN SYSTEMS PERFORMANCE GROUP                |
|                   All Rights Reserved                        |
+==============================================================+
| FILENAME
|    plnew.c
| DESCRIPTION
|    OCI version (using PL/SQL stored procedure) of
|    NEW ORDER transaction in TPC-C benchmark.
+==============================================================*/

#ifndef ORA_TPCC
# define ORA_TPCC
# include "tpcc.h"
#endif

#ifdef TUX
#include <userlog.h>
#endif

#ifndef ORA_TPCCFLAGS
# define ORA_TPCCFLAGS
# include "tpccflags.h"
#endif

#include "plora.h"

#define SQLTXT2 "BEGIN inittpcc.init_no(:idx1arr);END;"

#define NITEMS 15
#define ROWIDLEN 20
#define OCIROWLEN 20

struct newctx {

 ub2 nol_i_id_len[NITEMS];
 ub2 nol_supply_w_id_len[NITEMS];
 ub2 nol_quantity_len[NITEMS];
 ub2 nol_amount_len[NITEMS];
 ub2 s_quantity_len[NITEMS];
 ub2 i_name_len[NITEMS];
 ub2 i_price_len[NITEMS];
 ub2 s_dist_info_len[NITEMS];
 ub2 ol_o_id_len[NITEMS];
 ub2 ol_number_len[NITEMS];
 ub2 s_remote_len[NITEMS];
 ub2 s_quant_len[NITEMS];
 ub2 ol_dist_info_len[NITEMS];
 ub2 s_bg_len[NITEMS];

 int ol_o_id[NITEMS];
 int ol_number[NITEMS];

 int s_remote[NITEMS];
 char s_dist_info[NITEMS][25];
 OCIStmt *curn1;
 OCIBind *ol_i_id_bp;
 OCIBind *ol_supply_w_id_bp;
 OCIBind *i_price_bp;
 OCIBind *i_name_bp;
 OCIBind *s_bg_bp;
 ub4 nol_i_count;
 ub4 nol_s_count;
 ub4 nol_q_count;
 ub4 nol_item_count;
 ub4 nol_name_count;
 ub4 nol_qty_count;
 ub4 nol_bg_count;
 ub4 nol_am_count;
 ub4 s_remote_count;
 OCIStmt *curn2;
 OCIBind *ol_quantity_bp;
 OCIBind *s_remote_bp;
 OCIBind *s_quantity_bp;
 OCIBind *w_id_bp;
 OCIBind *d_id_bp;
 OCIBind *c_id_bp;
 OCIBind *o_all_local_bp;
 OCIBind *o_all_cnt_bp;
 OCIBind *w_tax_bp;
 OCIBind *d_tax_bp;
 OCIBind *o_id_bp;
 OCIBind *c_discount_bp;
 OCIBind *c_credit_bp;
 OCIBind *c_last_bp;
 OCIBind *retries_bp;
 OCIBind *cr_date_bp;
 OCIBind *ol_o_id_bp;
 OCIBind *ol_amount_bp;

 ub2 w_id_len; /* RP - may need to be (ub2), was (sb2) */
 ub2 d_id_len;
 ub2 c_id_len;
 ub2 o_all_local_len;
 ub2 o_ol_cnt_len;
 ub2 w_tax_len;
 ub2 d_tax_len;
 ub2 o_id_len;
 ub2 c_discount_len;
 ub2 c_credit_len;
 ub2 c_last_len;
 ub2 retries_len;
 ub2 cr_date_len;
 };

 typedef struct newctx newctx;

 /* static newctx *nctx; */

 tkvcninit (ora_cn_data_t *ora_SlotDataP)
 {

  int i;
  text stmbuf[32*1024];

  newctx *nctx;
  OCIEnv *tpcenv = ora_SlotDataP->tpcenv;
  OCIServer *tpcsrv = ora_SlotDataP->tpcsrv;
  OCIError *errhp = ora_SlotDataP->errhp;
  OCISvcCtx *tpcsvc = ora_SlotDataP->tpcsvc;
  OCISession *tpcusr = ora_SlotDataP->tpcusr;
  OCIStmt *curi = ora_SlotDataP->curi;
  global_newOrder_t *newP;

  nctx = (newctx *) malloc (sizeof(newctx));
  DISCARD memset(nctx,(char)0,sizeof(newctx));
  ora_SlotDataP->nctx = (void *)nctx;

  ora_SlotDataP->globals = (global_newOrder_t *)malloc(sizeof(global_newOrder_t));
  memset(ora_SlotDataP->globals,(char)0,sizeof(global_newOrder_t));
  newP = ora_SlotDataP->globals;

  nctx->w_id_len = sizeof(newP->w_id);
  nctx->d_id_len = sizeof(newP->d_id);
  nctx->c_id_len = sizeof(newP->c_id);
  nctx->o_all_local_len= sizeof(newP->o_all_local);
  nctx->o_ol_cnt_len = sizeof(newP->o_ol_cnt);
  nctx->w_tax_len = 0;
  nctx->d_tax_len = 0;
  nctx->o_id_len = sizeof(newP->o_id);
  nctx->c_discount_len = 0;
  nctx->c_credit_len = 0;
  nctx->c_last_len = 0;
  nctx->retries_len = sizeof(newP->retries);
  nctx->cr_date_len = sizeof(newP->cr_date);

  /* open first cursor */
  DISCARD OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid**)(&nctx->curn1),
        OCI_HTYPE_STMT, 0, (dvoid**)0));
 #if defined(ISO)
  sqlfile("tkvcpnew_iso.sql",stmbuf);
 #else
 #if defined(ISO7)
  sqlfile("tkvcpnew_iso7.sql",stmbuf);
 #else
  sqlfile("tkvcpnew.sql",stmbuf);
 #endif
 #endif

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

  /* bind variables */

  OCIBNDPL(nctx->curn1, nctx->w_id_bp, errhp, ":w_id",ADR(newP->w_id),SIZ(newP->w_id),
        SQLT_INT, &nctx->w_id_len);
  OCIBNDPL(nctx->curn1, nctx->d_id_bp, errhp, ":d_id",ADR(newP->d_id),SIZ(newP->d_id),
        SQLT_INT, &nctx->d_id_len);
```

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

OCIBNDPLA(nctx->curn1, nctx->ol_i_id_bp,errhp,":ol_i_id",newP->nol_i_id,
         SIZ(int), SQLT_INT, nctx->nol_i_id_len,NITEMS,&nctx->nol_i_count);
OCIBNDPLA(nctx->curn1, nctx->ol_supply_w_id_bp,errhp, ":ol_supply_w_id",
        newP->nol_supply_w_id,SIZ(int),SQLT_INT, nctx->nol_supply_w_id_len,
        NITEMS, &nctx->nol_s_count);
OCIBNDPLA(nctx->curn1, nctx->ol_quantity_bp,errhp,":ol_quantity",
        newP->nol_quantity, SIZ(int),SQLT_INT,nctx->nol_quantity_len,
        NITEMS,&nctx->nol_q_count);
OCIBNDPLA(nctx->curn1, nctx->i_price_bp,errhp,":i_price",newP->i_price,SIZ(float),
        SQLT_FLT, nctx->i_price_len, NITEMS, &nctx->nol_item_count);
OCIBNDPLA(nctx->curn1, nctx->i_name_bp,errhp,":i_name",newP->i_name,
        SIZ(newP->i_name[0]),SQLT_STR, nctx->i_name_len,NITEMS,
        &nctx->nol_name_count);
OCIBNDPLA(nctx->curn1, nctx->s_quantity_bp,errhp,":s_quantity",newP->s_quantity,
        SIZ(int), SQLT_INT,nctx->s_quant_len,NITEMS,&nctx->nol_qty_count);
OCIBNDPLA(nctx->curn1, nctx->s_bg_bp,errhp,":brand_generic",newP->brand_generic,
        SIZ(char), SQLT_CHR,nctx->s_bg_len,NITEMS,&nctx->nol_bg_count);
OCIBNDPLA(nctx->curn1, nctx->ol_amount_bp,errhp,":ol_amount",newP->nol_amount,
        SIZ(int),SQLT_INT, nctx->nol_amount_len,NITEMS,&nctx->nol_am_count);
OCIBNDPLA(nctx->curn1, nctx->s_remote_bp,errhp,":s_remote",nctx->s_remote,
        SIZ(int),SQLT_INT, nctx->s_remote_len,NITEMS,&nctx->s_remote_count);

/* open second cursor */
DISCARD OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid **)(&nctx->curn2),
              OCI_HTYPE_STMT, 0, (dvoid**)0));
DISCARD sprintf ((char *) stmbuf, SQLTXT2);
DISCARD OCIERROR(errhp,OCIStmtPrepare(nctx->curn2, errhp, stmbuf,
              strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));


/* execute second cursor to init newinit package */
{
 int idx1arr[NITEMS];
 OCIBind *idx1arr_bp;
 ub2 idx1arr_len[NITEMS];
 ub2 idx1arr_rcode[NITEMS];
 sb2 idx1arr_ind[NITEMS];
 ub4 idx1arr_count;
 ub2 idx;

 for (idx = 0; idx < NITEMS; idx++) {
   idx1arr[idx] = idx + 1;
   idx1arr_ind[idx] = TRUE;
   idx1arr_len[idx] = sizeof(int);
 }
 idx1arr_count = NITEMS;
                 newP->o_ol_cnt = NITEMS;


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

  newP->execstatus = OCIStmtExecute(tpcsvc,nctx->curn2,errhp,1,0,
            NULLP(CONST OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);
  if(newP->execstatus != OCI_SUCCESS) {
    OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
    newP->errcode = OCIERROR(errhp,newP->execstatus);
    return -1;
  }
 }

return (0);

}



tkvcn (ora_cn_data_t *ora_SlotDataP)
{

 int i;
 int rcount;

 OCIEnv *tpcenv = ora_SlotDataP->tpcenv;
 OCIServer *tpcsrv = ora_SlotDataP->tpcsrv;
 OCIError *errhp = ora_SlotDataP->errhp;
 OCISvcCtx *tpcsvc = ora_SlotDataP->tpcsvc;
 OCISession *tpcusr = ora_SlotDataP->tpcusr;
 OCIStmt *curi = ora_SlotDataP->curi;
 global_newOrder_t *newP = ora_SlotDataP->globals;
 newctx *nctx = (newctx *)ora_SlotDataP->nctx;

retry:

 newP->status = 0;                    /* number of invalid items*/

 /* get number of order lines, and check if all are local*/

 newP->o_ol_cnt = NITEMS;
 newP->o_all_local = 1;
 for (i = 0; i < NITEMS; i++) {
   if (newP->nol_i_id[i] == 0) {
     newP->o_ol_cnt = i;
     break;
   }
   if (newP->nol_supply_w_id[i] != newP->w_id) {
     nctx->s_remote[i] = 1;
     newP->o_all_local = 0;
   }
   else
     nctx->s_remote[i] = 0;
 }

 nctx->w_id_len = sizeof(newP->w_id);
 nctx->d_id_len = sizeof(newP->d_id);
 nctx->c_id_len = sizeof(newP->c_id);
 nctx->o_all_local_len = sizeof(newP->o_all_local);
 nctx->o_ol_cnt_len = sizeof(newP->o_ol_cnt);
 nctx->w_tax_len = 0;
 nctx->d_tax_len = 0;
 nctx->o_id_len = sizeof(newP->o_id);
 nctx->c_discount_len = 0;
 nctx->c_credit_len = 0;
 nctx->c_last_len = 0;
 nctx->retries_len = sizeof(newP->retries);
 nctx->cr_date_len = sizeof(newP->cr_date);
 /* this is the row count */
 rcount = newP->o_ol_cnt;
 nctx->nol_i_count = newP->o_ol_cnt;
 nctx->nol_q_count = newP->o_ol_cnt;
 nctx->nol_s_count = newP->o_ol_cnt;
 nctx->s_remote_count = newP->o_ol_cnt;

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

 /* initialization for array operations*/
 for (i = 0; i < newP->o_ol_cnt; i++) {
   nctx->ol_number[i] = i + 1;
   nctx->nol_i_id_len[i] = sizeof(int);
   nctx->nol_supply_w_id_len[i] = sizeof(int);
   nctx->nol_quantity_len[i] = sizeof(int);
   nctx->nol_amount_len[i] = sizeof(int);
   nctx->ol_o_id_len[i] = sizeof(int);
   nctx->ol_number_len[i] = sizeof(int);
   nctx->ol_dist_info_len[i] = nctx->s_dist_info_len[i];
   nctx->s_remote_len[i] = sizeof(int);
   nctx->s_quant_len[i] = sizeof(int);
   nctx->i_name_len[i]=0;
   nctx->s_bg_len[i] = 0;
 }
 for (i = newP->o_ol_cnt; i < NITEMS; i++) {

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

 newP->execstatus = OCIStmtExecute(tpcsvc,nctx->curn1,errhp,1,0,0,0,
                                          OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);


 if(newP->execstatus != OCI_SUCCESS) {
   OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
   newP->errcode = OCIERROR(errhp,newP->execstatus);
   if(newP->errcode == NOT_SERIALIZABLE) {
              newP->retries++;
              goto retry;
   } else if (newP->errcode == RECOVERR) {
              newP->retries++;
              goto retry;
   } else if (newP->errcode == SNAPSHOT_TOO_OLD) {
              newP->retries++;
```

```
                goto retry;
  } else {
                return -1;
  }
 }

 /* did the txn succeed ? */
 if (rcount != newP->o_ol_cnt)
 {
  newP->status = rcount - newP->o_ol_cnt;
  newP->o_ol_cnt = rcount;
 }

#ifdef DEBUG
 printf("w_id = %d, d_id = %d, c_id = %d\n",w_id, d_id, c_id);
#endif

 newP->total_amount = 0;
 for (i = 0; i < newP->o_ol_cnt; i++)
 {
      newP->total_amount += newP->nol_amount[i];
 }
 newP->total_amount *= ((float)(1 - newP->c_discount)) * (float)(1.0 + ((float)(newP->d_tax)) +
((float)(newP->w_tax)));
 newP->total_amount = newP->total_amount/100;

 return (0);

}



void tkvcndone (ora_cn_data_t *ora_SlotDataP)

{

 int i;

 newctx *nctx = (newctx *)ora_SlotDataP->nctx;
 global_newOrder_t *newP = ora_SlotDataP->globals;


 if (nctx)
 {
  DISCARD OCIHandleFree((dvoid *)nctx->curn1,OCI_HTYPE_STMT);
  DISCARD OCIHandleFree((dvoid *)nctx->curn2,OCI_HTYPE_STMT);
  free (nctx);
 }
 if (newP) {
  err_printf("free_handles>newP: 0x%x\n", newP);
  free(newP);
  ora_SlotDataP->globals = NULL;
 }
}
```

plora.h

```
/*
* $Header: plora.h for Encina
*/
/*+=================================================================+
|                    All Rights Reserved                          |
+=================================================================+
| FILENAME
|    plora.h
| DESCRIPTION
|    Include file for TPC-C benchmark programs.
+=================================================================*/


#ifndef TPCC_PLORA_H
#define TPCC_PLORA_H

#include "tpcc.h"
#include "tpcc_info.h"

#define NEWO_TRANS (1)
#define PAYMENT_TRANS (2)
#define ORDER_STAT_TRANS (3)
#define DELIVERY_TRANS (4)
#define STOCK_TRANS (5)
#define MAX_TRAN_TYPE (5)

/* struct to copy-in/out neworder vars */
struct global_newOrder_t {
 int w_id;
 int d_id;
 int c_id;
 int nol_i_id[15];
 int nol_supply_w_id[15];
 int nol_quantity[15];
 int retries;
 ub4  datelen;
 text o_entry_d[20];
 int o_id;
 int o_ol_cnt;
 char c_last[17];
 char c_credit[3];
 float c_discount;
 float w_tax;
 float d_tax;
 float total_amount;
 char i_name[15][25];
 int s_quantity[15];
 char brand_gen[15];
 float i_price[15];
 int nol_amount[15];
 int status;
 int o_all_local;
 int errcode;
 int execstatus;
 int proc_no;
 char brand_generic[15][1];
 int tracelevel;
 OCIDate cr_date;
 OCIDate c_since;
 OCIDate o_entry_d_base;
 OCIDate ol_d_base[15];

};

typedef struct global_newOrder_t global_newOrder_t;

struct global_payment_t {
 int w_id;
 int d_id;
 int c_id;
 char c_last[17];
 char c_first[17];
 char c_middle[3];
 double c_balance;
 int retries;
 int bylastname;
 OCIDate c_since;
 int execstatus;
 int errcode;
 int c_w_id;
 int c_d_id;
 int h_amount;
 char w_street_1[21];
 char w_street_2[21];
 char w_city[21];
 char w_state[3];
 char w_zip[10];
 char d_street_1[21];
 char d_street_2[21];
 char d_city[21];
 char d_state[3];
 char d_zip[10];
 char c_street_1[21];
 char c_street_2[21];
 char c_city[21];
 char c_state[3];
 char c_zip[10];
 char c_phone[17];
 ub4  sincelen;
 text c_since_d[11];
 float c_discount;
 char c_credit[3];
 int c_credit_lim;
 char c_data[201];
 ub4 hlen;
 text h_date[20];
 OCIDate cr_date;
};

typedef struct global_payment_t global_payment_t;

struct global_order_t {
 OCIDate ol_d_base[15];
 int w_id;
 int d_id;
 int c_id;
 char c_last[17];
 char c_first[17];
 char c_middle[3];
 double c_balance;
 int o_id;
 int o_carrier_id;
 int o_ol_cnt;
 int ol_supply_w_id[15];
 int ol_i_id[15];
 unsigned char o_entry_d_base[7];
 int ol_quantity[15];
 ub4  ol_del_len[15];
 text ol_delivery_d[15][11];
 int ol_amount[15];
 int errcode;
 int execstatus;
 int retries;
 int bylastname;
 char o_entry_d[20];
};

typedef struct global_order_t global_order_t;
```

```
struct global_delivery_t{
 int w_id;
 int o_carrier_id;
 int retries;
 int del_o_id[10];
 int errcode;
 int execstatus;
 int proc_no;
 OCIDate cr_date;
 int plsqlflag;
};
typedef struct global_delivery_t global_delivery_t;

struct global_stock_t {
 int w_id;
 int d_id;
 int threshold;
 int retries;
 int low_stock;
 int errcode;
 int execstatus;
};

typedef struct global_stock_t global_stock_t;


/* Oracle handles and rest of thread specificvars(thread slot data ) */

struct ora_cn_data_t {
 OCIEnv *tpcenv;
 OCIServer *tpcsrv;
 OCIError *errhp;
 OCISvcCtx *tpcsvc;
 OCISession *tpcusr;
 OCIStmt *curi;
 dvoid *xmem;

 global_newOrder_t *globals;
 global_payment_t *payP;
 global_order_t *ordP;
 global_delivery_t*delP;
 global_stock_t *stoP;
 void *nctx;
 void *pctx;
 void *octx;
 void *sctx;
 void *dctx;
 void *pldctx;
 void *actx;  /* for #ifdef DMLRETDEL */
 void *cbctx; /* for orderstatus */
 void *ctxp_octx; /* for orderstatus */
};

typedef struct ora_cn_data_t ora_cn_data_t;

#endif /* TPCC_PLORA_H */
```

plord.c

```
 /* Copyright (c) 2002, Oracle Corporation.  All rights reserved.  */

/*

 NAME
   tkvcordq.c - OCI version  using queues of ORDER STATUS
                transaction in TPC-C benchmark.

 DESCRIPTION
   <short description of facility this file declares/defines>

 EXPORT FUNCTION(S)

 INTERNAL FUNCTION(S)
   <other external functions defined - one-line descriptions>

 STATIC FUNCTION(S)
   <static functions defined - one-line descriptions>

 NOTES
   <other useful comments, qualifications,etc.>

 MODIFIED   (MM/DD/YY)
 xnie       06/25/02 - queue open cluster join.
 heri       05/07/02 - Fix error in cursor.
 heri       02/01/02 - Cleanup, remove indicator values and return codes.
 lwang      07/25/01 - Merged lwang_tpccitrc
 lwang      07/23/01 - fix include
 lwang      07/23/01 - Creation

*/


#include "tpcc.h"
#include "plora.h"
#include "tpccflags.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(ordl) CLUSTER(ordl) */ \
                ol_i_id, ol_supply_w_id, ol_quantity, ol_amount, ol_delivery_d\
                FROM ordr, ordl \
                WHERE ordr.rowid = :ordr_rowid \
                 AND  o_id = ol_o_id AND  ol_d_id = o_d_id AND ol_w_id = o_w_id"

#define SQLCUR4 "SELECT count(c_last) FROM cust \
                 WHERE  c_d_id = :d_id AND c_w_id = :w_id AND c_last = :c_last "


struct ordctx {

  ub2 c_rowid_len[100];
  ub2 ol_supply_w_id_len[NITEMS];
  ub2 ol_i_id_len[NITEMS];
  ub2 ol_quantity_len[NITEMS];
  ub2 ol_amount_len[NITEMS];
  ub2 ol_delivery_d_len[NITEMS];
  ub2 ol_w_id_len;
  ub2 ol_d_id_len;
  ub2 ol_o_id_len;


  ub4 ol_supply_w_id_csize;
  ub4 ol_i_id_csize;
  ub4 ol_quantity_csize;
  ub4 ol_amount_csize;
  ub4 ol_delivery_d_csize;
  ub4 ol_w_id_csize;
  ub4 ol_d_id_csize;
  ub4 ol_o_id_csize;

  OCIStmt *curo0;
  OCIStmt *curo1;
  OCIStmt *curo2;
  OCIStmt *curo3;
  OCIStmt *curo4;
  OCIBind *c_id_bp;
  OCIBind *w_id_bp[4];
  OCIBind *d_id_bp[4];
  OCIBind *c_last_bp[2];
  OCIBind *o_id_bp;
  OCIBind *c_rowid_bp;
  OCIBind *o_rowid_bp;
  OCIDefine *c_rowid_dp;
  OCIDefine *c_last_dp[2];
  OCIDefine *c_id_dp;
  OCIDefine *c_first_dp[2];
  OCIDefine *c_middle_dp[2];
  OCIDefine *c_balance_dp[2];
  OCIDefine *o_rowid_dp[2];
  OCIDefine *o_id_dp[2];
  OCIDefine *o_entry_d_dp[2];
  OCIDefine *o_cr_id_dp[2];
  OCIDefine *o_ol_cnt_dp[2];
  OCIDefine *ol_d_d_dp;
  OCIDefine *ol_i_id_dp;
  OCIDefine *ol_supply_w_id_dp;
  OCIDefine *ol_quantity_dp;
  OCIDefine *ol_amount_dp;
  OCIDefine *ol_d_base_dp;
  OCIDefine *c_count_dp;
  OCIRowid *c_rowid_ptr[100];
  OCIRowid *c_rowid_cust;
```

```
 OCIRowid *o_rowid;
 int cs;
 int cust_idx;
 int norow;
 int rcount;
 int somerows;
};

typedef struct ordctx ordctx;

struct defctx
{
boolean reexec;
ub4 count;
};
typedef struct defctx defctx;

struct defctx_ordctx {
  defctx *ctxp;
  ordctx *octx;
};
typedef struct defctx_ordctx defctx_ordctx;

/* static ordctx *octx; */

/* static defctx cbctx; */


tkvcoinit (ora_cn_data_t *ora_SlotDataP)

{

 int i;
 text stmbuf[SQL_BUF_SIZE];

 ordctx *octx;
 defctx *cbctx;
 defctx_ordctx *ctxp_octx;
 global_order_t *ordP;
 OCIEnv *tpcenv = ora_SlotDataP->tpcenv;
 OCIServer *tpcsrv = ora_SlotDataP->tpcsrv;
 OCIError *errhp = ora_SlotDataP->errhp;
 OCISvcCtx *tpcsvc = ora_SlotDataP->tpcsvc;
 OCISession *tpcusr = ora_SlotDataP->tpcusr;
 OCIStmt *curi = ora_SlotDataP->curi;

 octx = (ordctx *) malloc (sizeof(ordctx));
 DISCARD memset(octx,(char)0,sizeof(ordctx));
 octx->cs = 1;
 octx->norow = 0;
 octx->somerows = 10;
 ora_SlotDataP->octx = (void *)octx;

 cbctx = (defctx *) malloc (sizeof(defctx));
 DISCARD memset(cbctx,(char)0,sizeof(defctx));
 ora_SlotDataP->cbctx = (void *)cbctx;

 /* allocate the space */
 ctxp_octx = (defctx_ordctx *)malloc(sizeof(defctx_ordctx));
 ora_SlotDataP->ctxp_octx = (void *)ctxp_octx;

 ora_SlotDataP->ordP = (global_order_t *)malloc(sizeof(global_order_t));
 memset(ora_SlotDataP->ordP,(char)0,sizeof(global_order_t));
 ordP = ora_SlotDataP->ordP;


 /* get the rowid handles */
 OCIERROR(errhp, OCIDescriptorAlloc((dvoid *)tpcenv,(dvoid **)&octx->o_rowid,
              (ub4)OCI_DTYPE_ROWID, (size_t) 0, (dvoid **)0));
 for(i=0;i<100;i++) {
  DISCARD OCIERROR(errhp, OCIDescriptorAlloc(tpcenv,
            (dvoid**)&octx->c_rowid_ptr[i], OCI_DTYPE_ROWID,0,(dvoid**)0));
 }


 DISCARD OCIERROR(errhp,
   OCIHandleAlloc(tpcenv,(dvoid**)&octx->curo0,OCI_HTYPE_STMT,0,(dvoid**)0));
 DISCARD OCIERROR(errhp,
   OCIHandleAlloc(tpcenv,(dvoid**)&octx->curo0,OCI_HTYPE_STMT,0,(dvoid**)0));
 DISCARD OCIERROR(errhp,
   OCIHandleAlloc(tpcenv,(dvoid**)&octx->curo1,OCI_HTYPE_STMT,0,(dvoid**)0));
 DISCARD OCIERROR(errhp,
   OCIHandleAlloc(tpcenv,(dvoid**)&octx->curo2,OCI_HTYPE_STMT,0,(dvoid**)0));
 DISCARD OCIERROR(errhp,
   OCIHandleAlloc(tpcenv,(dvoid**)&octx->curo3,OCI_HTYPE_STMT,0,(dvoid**)0));
 DISCARD OCIERROR(errhp,
   OCIHandleAlloc(tpcenv,(dvoid**)&octx->curo4,OCI_HTYPE_STMT,0,(dvoid**)0));

/*  c_id = 0, use find customer by lastname. Get an array or rowid's back*/
 DISCARD sprintf((char *) stmbuf, SQLCUR0);
 DISCARD OCIERROR(errhp,
   OCIStmtPrepare(octx->curo0,errhp,stmbuf,(ub4)strlen((char *)stmbuf),
                  OCI_NTV_SYNTAX,OCI_DEFAULT));
 DISCARD OCIERROR(errhp,
   OCIAttrSet(octx->curo0,OCI_HTYPE_STMT,&octx->norow,0,
          OCI_ATTR_PREFETCH_ROWS,errhp));
/* get order/customer info back based on rowid */
 DISCARD sprintf((char *) stmbuf, SQLCUR1);
 DISCARD OCIERROR(errhp,
   OCIStmtPrepare(octx->curo1,errhp,stmbuf,(ub4)strlen((char *)stmbuf),
          OCI_NTV_SYNTAX,OCI_DEFAULT));
 DISCARD OCIERROR(errhp,
   OCIAttrSet(octx->curo1,OCI_HTYPE_STMT,&octx->norow,0,
          OCI_ATTR_PREFETCH_ROWS,errhp));

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

 DISCARD sprintf((char *) stmbuf, SQLCUR3);
 DISCARD OCIERROR(errhp,
   OCIStmtPrepare(octx->curo3,errhp,stmbuf,(ub4)strlen((char *)stmbuf),
          OCI_NTV_SYNTAX,OCI_DEFAULT));
 DISCARD OCIERROR(errhp,
   OCIAttrSet(octx->curo3,OCI_HTYPE_STMT,&octx->norow,0,
          OCI_ATTR_PREFETCH_ROWS,errhp));

 DISCARD sprintf((char *) stmbuf, SQLCUR4);
 DISCARD OCIERROR(errhp,
   OCIStmtPrepare(octx->curo4,errhp,stmbuf,(ub4)strlen((char *)stmbuf),
          OCI_NTV_SYNTAX,OCI_DEFAULT));
 DISCARD OCIERROR(errhp,
   OCIAttrSet(octx->curo4,OCI_HTYPE_STMT,&octx->norow,0,
          OCI_ATTR_PREFETCH_ROWS,errhp));

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

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

 /* bind variables*/

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

 OCIBND(octx->curo1,octx->c_rowid_bp,errhp,":cust_rowid", &octx->c_rowid_cust,
      sizeof( octx->c_rowid_ptr[0]),SQLT_RDD);
 OCIDEF(octx->curo1,octx->c_id_dp,errhp,1,ADR(ordP->c_id),SIZ(int),SQLT_INT);
 OCIDEF(octx->curo1,octx->c_balance_dp[0],errhp,2,ADR(ordP->c_balance),
      SIZ(double),SQLT_FLT);
 OCIDEF(octx->curo1,octx->c_first_dp[0],errhp,3,ordP->c_first,SIZ(ordP->c_first)-1,
      SQLT_CHR);
 OCIDEF(octx->curo1,octx->c_middle_dp[0],errhp,4,ordP->c_middle,
      SIZ(ordP->c_middle)-1,SQLT_AFC);
 OCIDEF(octx->curo1,octx->c_last_dp[0],errhp,5,ordP->c_last,SIZ(ordP->c_last)-1,
      SQLT_CHR);
 OCIDEF(octx->curo1,octx->o_id_dp[0],errhp,6,ADR(ordP->o_id),SIZ(int),SQLT_INT);
 OCIDEF(octx->curo1,octx->o_entry_d_dp[0],errhp,7,
      &ordP->o_entry_d_base,SIZ(OCIDate),SQLT_ODT);
 OCIDEF(octx->curo1,octx->o_cr_id_dp[0],errhp,8,ADR(ordP->o_carrier_id),
      SIZ(int),SQLT_INT);
 OCIDEF(octx->curo1,octx->o_ol_cnt_dp[0],errhp,9,ADR(ordP->o_ol_cnt),
      SIZ(int),SQLT_INT);
 OCIDEF(octx->curo1,octx->o_rowid_dp[0],errhp,10,ADR(octx->o_rowid),
      SIZ(OCIRowid*),SQLT_RDD);


/* Bind for third cursor , no-zero customer id */
 OCIBND(octx->curo2,octx->w_id_bp[1],errhp,":w_id",ADR(ordP->w_id),
      SIZ(int),SQLT_INT);
 OCIBND(octx->curo2,octx->d_id_bp[1],errhp,":d_id",ADR(ordP->d_id),
                SIZ(int),SQLT_INT);
 OCIBND(octx->curo2,octx->c_id_bp,errhp,":c_id",ADR(ordP->c_id),
                SIZ(int),SQLT_INT);
 OCIDEF(octx->curo2,octx->c_balance_dp[1],errhp,1,ADR(ordP->c_balance),
                SIZ(double),SQLT_FLT);
 OCIDEF(octx->curo2,octx->c_first_dp[1],errhp,2,ordP->c_first,SIZ(ordP->c_first)-1,
      SQLT_CHR);
 OCIDEF(octx->curo2,octx->c_middle_dp[1],errhp,3,ordP->c_middle,
      SIZ(ordP->c_middle)-1,SQLT_AFC);
 OCIDEF(octx->curo2,octx->c_last_dp[1],errhp,4,ordP->c_last,SIZ(ordP->c_last)-1,
```

```
      SQLT_CHR);
 OCIDEF(octx->curo2,octx->o_id_dp[1],errhp,5,ADR(ordP->o_id),SIZ(int),SQLT_INT);
 OCIDEF(octx->curo2,octx->o_entry_d_dp[1],errhp,6, &ordP->o_entry_d_base,
      SIZ(OCIDate),SQLT_ODT);
 OCIDEF(octx->curo2, octx->o_cr_id_dp[1],errhp,7,ADR(ordP->o_carrier_id),
       SIZ(int), SQLT_INT);
 OCIDEF(octx->curo2,octx->o_ol_cnt_dp[1],errhp,8,ADR(ordP->o_ol_cnt),
       SIZ(int),SQLT_INT);
 OCIDEF(octx->curo2,octx->o_rowid_dp[1],errhp,9,ADR(octx->o_rowid),
      SIZ(OCIRowid*),SQLT_RDD);

/* Bind for last cursor */

/*
 OCIBND(octx->curo3,octx->w_id_bp[2],errhp,":w_id",ADR(ordP->w_id), SIZ(int),SQLT_INT);
 OCIBND(octx->curo3,octx->d_id_bp[2],errhp,":d_id",ADR(ordP->d_id), SIZ(int),SQLT_INT);
 OCIBND(octx->curo3,octx->o_id_bp,errhp,":o_id",ADR(ordP->o_id), SIZ(int),SQLT_INT);
 OCIBND(octx->curo3,octx->c_id_bp,errhp,":c_id",ADR(ordP->c_id), SIZ(int),SQLT_INT);
*/
 OCIBND(octx->curo3,octx->o_rowid_bp,errhp,":ordr_rowid",
      &octx->o_rowid, SIZ(OCIRowid*),SQLT_RDD);

 OCIDFNRA(octx->curo3, octx->ol_i_id_dp, errhp, 1, ordP->ol_i_id,SIZ(int),SQLT_INT,
                  NULL,octx->ol_i_id_len,NULL);
 OCIDFNRA(octx->curo3,octx->ol_supply_w_id_dp,errhp,2, ordP->ol_supply_w_id,
        SIZ(int),SQLT_INT, NULL,
        octx->ol_supply_w_id_len,NULL);
 OCIDFNRA(octx->curo3, octx->ol_quantity_dp,errhp,3, ordP->ol_quantity,SIZ(int),
        SQLT_INT, NULL,octx->ol_quantity_len,NULL);
 OCIDFNRA(octx->curo3,octx->ol_amount_dp,errhp,4,ordP->ol_amount, SIZ(int),
        SQLT_INT,NULL, octx->ol_amount_len, NULL);
 OCIDFNRA(octx->curo3,octx->ol_d_base_dp,errhp,5,ordP->ol_d_base,SIZ(OCIDate),
        SQLT_ODT, NULL,octx->ol_delivery_d_len,NULL);

 OCIBND(octx->curo4,octx->w_id_bp[3],errhp,":w_id",ADR(ordP->w_id),
      SIZ(int),SQLT_INT);
 OCIBND(octx->curo4,octx->d_id_bp[3],errhp,":d_id",ADR(ordP->d_id),
      SIZ(int),SQLT_INT);
 OCIBND(octx->curo4,octx->c_last_bp[1],errhp,":c_last",ordP->c_last,
      SIZ(ordP->c_last), SQLT_STR);
 OCIDEF(octx->curo4,octx->c_count_dp,errhp,1,ADR(octx->rcount),SIZ(int),
      SQLT_INT);


 return (0);

}



tkvco (ora_cn_data_t *ora_SlotDataP)

{

 int i;
 int rcount;

 ordctx *octx = (ordctx *)ora_SlotDataP->octx;
 defctx *cbctx = (defctx *)ora_SlotDataP->cbctx;
 global_order_t *ordP = ora_SlotDataP->ordP;
 OCIEnv *tpcenv = ora_SlotDataP->tpcenv;
 OCIServer *tpcsrv = ora_SlotDataP->tpcsrv;
 OCIError *errhp = ora_SlotDataP->errhp;
 OCISvcCtx *tpcsvc = ora_SlotDataP->tpcsvc;
 OCISession *tpcusr = ora_SlotDataP->tpcusr;
 OCIStmt *curi = ora_SlotDataP->curi;


#if defined(ISO9)
 int secondread = 0;
 char sdate[30];
 ub4  datelen;
 sysdate(sdate);
 printf("Order Status started at: %s\n", sdate);
#endif

 for (i = 0; i < NITEMS; i++) {
   octx->ol_supply_w_id_len[i]= sizeof(int);
   octx->ol_i_id_len[i]= sizeof(int);
   octx->ol_quantity_len[i]= sizeof(int);
   octx->ol_amount_len[i] = sizeof(int);
   octx->ol_delivery_d_len[i]= sizeof(OCIDate);
 }
 octx->ol_supply_w_id_csize= NITEMS;
 octx->ol_i_id_csize= NITEMS;
 octx->ol_quantity_csize= NITEMS;
 octx->ol_amount_csize= NITEMS;
 octx->ol_delivery_d_csize= NITEMS;
retry:
 if(ordP->bylastname)
 {
  cbctx->reexec = FALSE;
  ordP->execstatus=OCIStmtExecute(tpcsvc,octx->curo0,errhp,100,0,
          NULLP(CONST OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);
  /* will get OCI_NO_DATA if <100 found */
  if ((ordP->execstatus != OCI_NO_DATA) && (ordP->execstatus != OCI_SUCCESS))
  {
    ordP->errcode=OCIERROR(errhp, ordP->execstatus);
    if((ordP->errcode == NOT_SERIALIZABLE) || (ordP->errcode == RECOVERR))
    {
     DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
     ordP->retries++;
     goto retry;
    } else {
     return -1;
    }
  }
  if (ordP->execstatus == OCI_NO_DATA) /* there are no more rows */
  {
   /* get rowcount, find middle one*/
   DISCARD OCIAttrGet(octx->curo0,OCI_HTYPE_STMT,&rcount,NULL,
              OCI_ATTR_ROW_COUNT,errhp);
   if (rcount <1)
   {
     userlog("ORDERSTATUS  rcount=%d\n",rcount);
     return (-1);
   }
   octx->cust_idx=(rcount)/2 ;
  }
  else
  {
    /* count the number of rows */
    ordP->execstatus=OCIStmtExecute(tpcsvc,octx->curo4,errhp,1,0,
            NULLP(CONST OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);
    if ((ordP->execstatus != OCI_NO_DATA) && (ordP->execstatus != OCI_SUCCESS))
    {
      ordP->errcode=OCIERROR(errhp, ordP->execstatus);
              if ((ordP->errcode == NOT_SERIALIZABLE) || (ordP->errcode == RECOVERR))
      {
       DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
       ordP->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 ;
      ordP->execstatus=OCIStmtExecute(tpcsvc,octx->curo0,errhp,cbctx->count,
                        0,NULLP(CONST OCISnapshot),
                        NULLP(OCISnapshot),OCI_DEFAULT);
      /* will get OCI_NO_DATA if <100 found */
      if (cbctx->count > 0)
      {
        userlog ("did not get all rows ");
        return (-1);
      }

      if ((ordP->execstatus != OCI_NO_DATA) && (ordP->execstatus != OCI_SUCCESS))
      {
        ordP->errcode=OCIERROR(errhp, ordP->execstatus);
        if((ordP->errcode == NOT_SERIALIZABLE) || (ordP->errcode == RECOVERR))
        {
         DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
         ordP->retries++;
         goto retry;
        } else {
        return -1;
        }
      }
      octx->cust_idx=0 ;
    }
  }

  octx->c_rowid_cust = octx->c_rowid_ptr[octx->cust_idx];
  ordP->execstatus=OCIStmtExecute(tpcsvc,octx->curo1,errhp,1,0,
              NULLP(CONST OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT);
 if (ordP->execstatus != OCI_SUCCESS)
 {
    ordP->errcode=OCIERROR(errhp,ordP->execstatus);
    DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
    if((ordP->errcode == NOT_SERIALIZABLE) || (ordP->errcode == RECOVERR)
      || (ordP->errcode == SNAPSHOT_TOO_OLD))
    {
      ordP->retries++;
      goto retry;
      } else {
      return -1;
    }
  }
 }
 else
 {
  ordP->execstatus=OCIStmtExecute(tpcsvc,octx->curo2,errhp,1,0,
                    NULLP(CONST OCISnapshot),NULLP(OCISnapshot),
                    OCI_DEFAULT);
  if (ordP->execstatus != OCI_SUCCESS)
  {
    ordP->errcode=OCIERROR(errhp,ordP->execstatus);
    DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
    if((ordP->errcode == NOT_SERIALIZABLE) || (ordP->errcode == RECOVERR)
      || (ordP->errcode == SNAPSHOT_TOO_OLD))
    {
      ordP->retries++;
```

```
      goto retry;
     }
     else
     {
      return -1;
     }
    }
#ifdef ISO9
    sysdate (sdate);
if (!secondread)
    printf ("---------- FIRST READ RESULT (out) %s ----------\n",  sdate);
else
    printf ("---------- SECOND READ RESULT (out) %s ----------\n",  sdate);

    printf ("c_id = %d\n", c_id);
    printf ("c_last = %s\n", c_last);
    printf ("c_first = %s\n", c_first);
    printf ("c_middle = %s\n", c_middle);
    printf ("c_balance = %7.2f\n", (float)c_balance/100);
    printf ("o_id = %d\n", o_id);
    datelen = sizeof(o_entry_d);

OCIERROR(errhp,OCIDateToText(errhp,&o_entry_d_base,(text*)FULLDATE,SIZ(FULLDATE),(text*
)0,0,&datelen,o_entry_d));
    printf ("o_entry_d = %s\n", o_entry_d);
    printf ("o_carrier_id = %d\n", o_carrier_id);
    printf ("o_ol_cnt = %d\n", o_ol_cnt);
    printf ("-------------------------------------------------\n\n",  sdate);

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

  ordP->execstatus = OCIStmtExecute(tpcsvc,octx->curo3,errhp,ordP->o_ol_cnt,0,
                      NULLP(CONST OCISnapshot),NULLP(OCISnapshot),
                      OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
  if (ordP->execstatus != OCI_SUCCESS )
  {
    ordP->errcode=OCIERROR(errhp,ordP->execstatus);
    DISCARD OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
    if((ordP->errcode == NOT_SERIALIZABLE) || (ordP->errcode == RECOVERR)
        || (ordP->errcode == SNAPSHOT_TOO_OLD))
    {
      ordP->retries++;
      goto retry;
    }
    else
    {
      return -1;
    }

  }
  /* clean up and convert the delivery dates */
  for (i = 0; i < ordP->o_ol_cnt; i++)
  {
    ordP->ol_del_len[i]=sizeof(ordP->ol_delivery_d[i]);
    DISCARD OCIERROR(errhp,OCIDateToText(errhp,&ordP->ol_d_base[i],
       (const text*)SHORTDATE,(ub1)strlen(SHORTDATE),(text*)0,0,
        &ordP->ol_del_len[i], ordP->ol_delivery_d[i]));
/*
                    cvtdmy(ol_d_base[i],ol_delivery_d[i]);
*/
  }

  return (0);

}



void tkvcodone (ora_cn_data_t *ora_SlotDataP)

{

  if (ora_SlotDataP->octx) {
    free (ora_SlotDataP->octx);
    ora_SlotDataP->octx = NULL;
  }
  if (ora_SlotDataP->ordP) {
    free(ora_SlotDataP->ordP);
    ora_SlotDataP->ordP = NULL;
  }
}

/* end of file tkvcord.c */
```

plpay.c

```
  #ifdef RCSID
static char *RCSid =
  "$Header: plpay.c 7030100.1 95/07/19 14:44:59 plai Generic<base> $ Copyr (c) 1994 Oracle for Encina";
#endif /* RCSID */

/*+============================================================+
|       Copyright (c) 1995  Oracle Corp, Redwood Shores, CA      |
|                OPEN SYSTEMS PERFORMANCE GROUP                  |
|                    All Rights Reserved                         |
+============================================================+
| FILENAME
|    plpay.c
| DESCRIPTION
|    OCI version (using PL/SQL stored procedure) of
|    PAYMENT transaction in TPC-C benchmark.
+============================================================*/

#include "tpcc.h"
#include "plora.h"
#include "tpccflags.h"

#ifdef TUX
#include <userlog.h>
#endif


#define SQLTXT_INIT "BEGIN inittpcc.init_pay;END;"

struct payctx {
 OCIStmt *curpi;
 OCIStmt *curp0;
 OCIStmt *curp1;
 OCIBind *w_id_bp[2];
 ub2 w_id_len;

 OCIBind *d_id_bp[2];
 ub2 d_id_len;

 OCIBind *c_w_id_bp[2];
 ub2 c_w_id_len;

 OCIBind *c_d_id_bp[2];
 ub2 c_d_id_len;

 OCIBind *c_id_bp[2];
 ub2 c_id_len;

 OCIBind *h_amount_bp[2];
 ub2 h_amount_len;

 OCIBind *c_last_bp[2];
 ub2 c_last_len;

 OCIBind *w_street_1_bp[2];
 ub2 w_street_1_len;

 OCIBind *w_street_2_bp[2];
 ub2 w_street_2_len;

 OCIBind *w_city_bp[2];
 ub2 w_city_len;

 OCIBind *w_state_bp[2];
 ub2 w_state_len;

 OCIBind *w_zip_bp[2];
 ub2 w_zip_len;

 OCIBind *d_street_1_bp[2];
 ub2 d_street_1_len;

 OCIBind *d_street_2_bp[2];
 ub2 d_street_2_len;

 OCIBind *d_city_bp[2];
 ub2 d_city_len;

 OCIBind *d_state_bp[2];
 ub2 d_state_len;

 OCIBind *d_zip_bp[2];
 ub2 d_zip_len;

 OCIBind *c_first_bp[2];
 ub2 c_first_len;

 OCIBind *c_middle_bp[2];
 ub2 c_middle_len;

 OCIBind *c_street_1_bp[2];
 ub2 c_street_1_len;

 OCIBind *c_street_2_bp[2];
```

```
ub2 c_street_2_len;

OCIBind *c_city_bp[2];
ub2 c_city_len;

OCIBind *c_state_bp[2];
ub2 c_state_len;

OCIBind *c_zip_bp[2];
ub2 c_zip_len;

OCIBind *c_phone_bp[2];
ub2 c_phone_len;

OCIBind *c_since_bp[2];
ub2 c_since_len;

OCIBind *c_credit_bp[2];
ub2 c_credit_len;

OCIBind *c_credit_lim_bp[2];
ub2 c_credit_lim_len;

OCIBind *c_discount_bp[2];
ub2 c_discount_len;

OCIBind *c_balance_bp[2];
ub2 c_balance_len;

OCIBind *c_data_bp[2];
ub2 c_data_len;

OCIBind *h_date_bp[2];
ub2 h_date_len;

OCIBind *retries_bp[2];
ub2 retries_len;

OCIBind *cr_date_bp[2];
ub2 cr_date_len;

OCIBind *byln_bp[2];
ub2 byln_len;
};

typedef struct payctx payctx;



/* payctx *pctx;  */

int tkvcpinit (ora_cn_data_t *ora_SlotDataP)

{

 /*  */
 char *ora_home = getenv("ORACLE_HOME");
 char sql_file_name[256];
 payctx *pctx;
 global_payment_t *payP;
 OCIEnv *tpcenv = ora_SlotDataP->tpcenv;
 OCIServer *tpcsrv = ora_SlotDataP->tpcsrv;
 OCIError *errhp = ora_SlotDataP->errhp;
 OCISvcCtx *tpcsvc = ora_SlotDataP->tpcsvc;
 OCISession *tpcusr = ora_SlotDataP->tpcusr;
 OCIStmt *curi = ora_SlotDataP->curi;

 text stmbuf[SQL_BUF_SIZE];

 if (!ora_home) {
   err_printf("Cannot find env variable ORACLE_HOME\n");
   exit(13);
 }

 pctx = (payctx *)malloc(sizeof(payctx));
 memset(pctx,(char)0,sizeof(payctx));
 ora_SlotDataP->pctx = (void *)pctx;

 ora_SlotDataP->payP = (global_payment_t *)malloc(sizeof(global_payment_t));
 memset(ora_SlotDataP->payP,(char)0,sizeof(global_payment_t));
 payP = ora_SlotDataP->payP;


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

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

 /*  build the init statement  and execute it */

 sprintf ((char*)stmbuf, SQLTXT_INIT);
 DISCARD OCIERROR(errhp,OCIStmtPrepare(pctx->curpi, errhp, stmbuf,
        strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));
 DISCARD OCIERROR(errhp, OCIStmtExecute(tpcsvc,pctx->curpi,errhp,1,0,
                NULLP(CONST OCISnapshot),NULLP(OCISnapshot),OCI_DEFAULT));

 /* customer id !=  0, go by last name */

 sqlfile("paynz.sql",stmbuf);
 DISCARD OCIERROR(errhp,OCIStmtPrepare(pctx->curp0, errhp, stmbuf,
        strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));

 /* customer id ==  0, go by last name */

 sqlfile("payz.sql",stmbuf);/* sqlfile opens $O/bench/.../blocks/... */
 DISCARD OCIERROR(errhp,OCIStmtPrepare(pctx->curp1, errhp, stmbuf,
        strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT));

 pctx->w_id_len = SIZ(payP->w_id);
 pctx->d_id_len = SIZ(payP->d_id);
 pctx->c_w_id_len = SIZ(payP->c_w_id);
 pctx->c_d_id_len = SIZ(payP->c_d_id);
 pctx->c_id_len = 0;
 pctx->h_amount_len = SIZ(payP->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(payP->retries) ;
 pctx->cr_date_len = 7;



 /* bind variables*/


 OCIBNDPL(pctx->curp0, pctx->w_id_bp[0], errhp,":w_id",ADR(payP->w_id),SIZ(int),
        SQLT_INT, NULL);
 OCIBNDPL(pctx->curp0, pctx->d_id_bp[0], errhp,":d_id",ADR(payP->d_id),SIZ(int),
        SQLT_INT, NULL);
 OCIBND(pctx->curp0, pctx->c_w_id_bp[0], errhp,":c_w_id",ADR(payP->c_w_id),SIZ(int),
        SQLT_INT);
 OCIBND(pctx->curp0, pctx->c_d_id_bp[0], errhp,":c_d_id",ADR(payP->c_d_id),SIZ(int),
        SQLT_INT);
 OCIBND(pctx->curp0, pctx->c_id_bp[0], errhp,":c_id",ADR(payP->c_id),SIZ(int),
        SQLT_INT);
 OCIBNDPL(pctx->curp0, pctx->h_amount_bp[0], errhp,":h_amount",ADR(payP->h_amount),
        SIZ(int),SQLT_INT,  &pctx->h_amount_len);
 OCIBNDPL(pctx->curp0, pctx->c_last_bp[0], errhp,":c_last",payP->c_last,SIZ(payP->c_last),
        SQLT_STR, &pctx->c_last_len);
 OCIBNDPL(pctx->curp0, pctx->w_street_1_bp[0], errhp,":w_street_1",payP->w_street_1,
        SIZ(payP->w_street_1),SQLT_STR, &pctx->w_street_1_len);
 OCIBNDPL(pctx->curp0, pctx->w_street_2_bp[0], errhp,":w_street_2",payP->w_street_2,
        SIZ(payP->w_street_2),SQLT_STR, &pctx->w_street_2_len);
 OCIBNDPL(pctx->curp0, pctx->w_city_bp[0], errhp,":w_city",payP->w_city,SIZ(payP->w_city),
        SQLT_STR, &pctx->w_city_len);
 OCIBNDPL(pctx->curp0, pctx->w_state_bp[0], errhp,":w_state",payP->w_state,
        SIZ(payP->w_state), SQLT_STR, &pctx->w_state_len);
 OCIBNDPL(pctx->curp0, pctx->w_zip_bp[0], errhp,":w_zip",payP->w_zip,SIZ(payP->w_zip),
        SQLT_STR, &pctx->w_zip_len);
 OCIBNDPL(pctx->curp0, pctx->d_street_1_bp[0], errhp,":d_street_1",payP->d_street_1,
        SIZ(payP->d_street_1),SQLT_STR, &pctx->d_street_1_len);
 OCIBNDPL(pctx->curp0, pctx->d_street_2_bp[0], errhp,":d_street_2",payP->d_street_2,
        SIZ(payP->d_street_2),SQLT_STR, &pctx->d_street_2_len);
 OCIBNDPL(pctx->curp0, pctx->d_city_bp[0], errhp,":d_city",payP->d_city,SIZ(payP->d_city),
        SQLT_STR, &pctx->d_city_len);
 OCIBNDPL(pctx->curp0, pctx->d_state_bp[0], errhp,":d_state",payP->d_state,
        SIZ(payP->d_state), SQLT_STR, &pctx->d_state_len);
 OCIBNDPL(pctx->curp0, pctx->d_zip_bp[0], errhp,":d_zip",payP->d_zip,SIZ(payP->d_zip),
        SQLT_STR, &pctx->d_zip_len);
 OCIBNDPL(pctx->curp0, pctx->c_first_bp[0], errhp,":c_first",payP->c_first,
        SIZ(payP->c_first), SQLT_STR, &pctx->c_first_len);
 OCIBNDPL(pctx->curp0, pctx->c_middle_bp[0], errhp,":c_middle",payP->c_middle,2,
        SQLT_AFC, &pctx->c_middle_len);
 OCIBNDPL(pctx->curp0, pctx->c_street_1_bp[0], errhp,":c_street_1",payP->c_street_1,
        SIZ(payP->c_street_1),SQLT_STR, &pctx->c_street_1_len);
 OCIBNDPL(pctx->curp0, pctx->c_street_2_bp[0], errhp,":c_street_2",payP->c_street_2,
        SIZ(payP->c_street_2),SQLT_STR, &pctx->c_street_2_len);
 OCIBNDPL(pctx->curp0, pctx->c_city_bp[0], errhp,":c_city",payP->c_city,SIZ(payP->c_city),
        SQLT_STR, &pctx->c_city_len);
 OCIBNDPL(pctx->curp0, pctx->c_state_bp[0], errhp,":c_state",payP->c_state,
```

```
      SIZ(payP->c_state), SQLT_STR, &pctx->c_state_len);
OCIBNDPL(pctx->curp0, pctx->c_zip_bp[0], errhp,":c_zip",payP->c_zip,SIZ(payP->c_zip),
      SQLT_STR,&pctx->c_zip_len);
OCIBNDPL(pctx->curp0, pctx->c_phone_bp[0], errhp,":c_phone",payP->c_phone,
      SIZ(payP->c_phone), SQLT_STR, &pctx->c_phone_len);
OCIBNDPL(pctx->curp0, pctx->c_since_bp[0], errhp,":c_since",&payP->c_since,
      SIZ(OCIDate), SQLT_ODT, &pctx->c_since_len);
OCIBNDPL(pctx->curp0, pctx->c_credit_bp[0], errhp,":c_credit",payP->c_credit,
      SIZ(payP->c_credit),SQLT_CHR, &pctx->c_credit_len);
OCIBNDPL(pctx->curp0, pctx->c_credit_lim_bp[0],errhp,":c_credit_lim",
      ADR(payP->c_credit_lim),SIZ(int),SQLT_INT, &pctx->c_credit_lim_len);
OCIBNDPL(pctx->curp0, pctx->c_discount_bp[0], errhp,":c_discount",
      ADR(payP->c_discount),SIZ(payP->c_discount), SQLT_FLT, &pctx->c_discount_len);
OCIBNDPL(pctx->curp0, pctx->c_balance_bp[0], errhp,":c_balance",
      ADR(payP->c_balance), SIZ(payP->c_balance),SQLT_FLT, &pctx->c_balance_len);
OCIBNDPL(pctx->curp0, pctx->c_data_bp[0], errhp,":c_data",payP->c_data,SIZ(payP->c_data),
      SQLT_STR, &pctx->c_data_len);
/*
OCIBNDR(pctx->curp0, pctx->h_date_bp, errhp,":h_date",h_date,SIZ(payP->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(payP->retries),
      SIZ(int), SQLT_INT, &pctx->retries_len);
OCIBNDPL(pctx->curp0, pctx->cr_date_bp[0], errhp,":cr_date",ADR(payP->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(payP->w_id),SIZ(int),
      SQLT_INT, &pctx->w_id_len);
OCIBNDPL(pctx->curp1, pctx->d_id_bp[1], errhp,":d_id",ADR(payP->d_id),SIZ(int),
      SQLT_INT, &pctx->d_id_len);
OCIBND(pctx->curp1, pctx->c_w_id_bp[1], errhp,":c_w_id",ADR(payP->c_w_id),SIZ(int),
      SQLT_INT);
OCIBND(pctx->curp1, pctx->c_d_id_bp[1], errhp,":c_d_id",ADR(payP->c_d_id),SIZ(int),
      SQLT_INT);
OCIBNDPL(pctx->curp1, pctx->c_id_bp[1], errhp,":c_id",ADR(payP->c_id),SIZ(int),
      SQLT_INT, &pctx->c_id_len);
OCIBNDPL(pctx->curp1, pctx->h_amount_bp[1], errhp,":h_amount",ADR(payP->h_amount),
      SIZ(int),SQLT_INT, &pctx->h_amount_len);
OCIBND(pctx->curp1, pctx->c_last_bp[1], errhp,":c_last",payP->c_last,SIZ(payP->c_last),
      SQLT_STR);
OCIBNDPL(pctx->curp1, pctx->w_street_1_bp[1], errhp,":w_street_1",payP->w_street_1,
      SIZ(payP->w_street_1),SQLT_STR, &pctx->w_street_1_len);
OCIBNDPL(pctx->curp1, pctx->w_street_2_bp[1], errhp,":w_street_2",payP->w_street_2,
      SIZ(payP->w_street_2),SQLT_STR, &pctx->w_street_2_len);
OCIBNDPL(pctx->curp1, pctx->w_city_bp[1], errhp,":w_city",payP->w_city,SIZ(payP->w_city),
      SQLT_STR, &pctx->w_city_len);
OCIBNDPL(pctx->curp1, pctx->w_state_bp[1], errhp,":w_state",payP->w_state,
      SIZ(payP->w_state), SQLT_STR, &pctx->w_state_len);
OCIBNDPL(pctx->curp1, pctx->w_zip_bp[1], errhp,":w_zip",payP->w_zip,SIZ(payP->w_zip),
      SQLT_STR, &pctx->w_zip_len);
OCIBNDPL(pctx->curp1, pctx->d_street_1_bp[1], errhp,":d_street_1",payP->d_street_1,
      SIZ(payP->d_street_1),SQLT_STR, &pctx->d_street_1_len);
OCIBNDPL(pctx->curp1, pctx->d_street_2_bp[1], errhp,":d_street_2",payP->d_street_2,
      SIZ(payP->d_street_2),SQLT_STR, &pctx->d_street_2_len);
OCIBNDPL(pctx->curp1, pctx->d_city_bp[1], errhp,":d_city",payP->d_city,SIZ(payP->d_city),
      SQLT_STR, &pctx->d_city_len);
OCIBNDPL(pctx->curp1, pctx->d_state_bp[1], errhp,":d_state",payP->d_state,
      SIZ(payP->d_state), SQLT_STR, &pctx->d_state_len);
OCIBNDPL(pctx->curp1, pctx->d_zip_bp[1], errhp,":d_zip",payP->d_zip,SIZ(payP->d_zip),
      SQLT_STR, &pctx->d_zip_len);
OCIBNDPL(pctx->curp1, pctx->c_first_bp[1], errhp,":c_first",payP->c_first,
      SIZ(payP->c_first), SQLT_STR, &pctx->c_first_len);
OCIBNDPL(pctx->curp1, pctx->c_middle_bp[1], errhp,":c_middle",payP->c_middle,2,
      SQLT_AFC, &pctx->c_middle_len);

OCIBNDPL(pctx->curp1, pctx->c_street_1_bp[1], errhp,":c_street_1",payP->c_street_1,
      SIZ(payP->c_street_1),SQLT_STR, &pctx->c_street_1_len);
OCIBNDPL(pctx->curp1, pctx->c_street_2_bp[1], errhp,":c_street_2",payP->c_street_2,
      SIZ(payP->c_street_2),SQLT_STR, &pctx->c_street_2_len);
OCIBNDPL(pctx->curp1, pctx->c_city_bp[1], errhp,":c_city",payP->c_city,
      SIZ(payP->c_city),SQLT_STR, &pctx->c_city_len);
OCIBNDPL(pctx->curp1, pctx->c_state_bp[1], errhp,":c_state",payP->c_state,
      SIZ(payP->c_state), SQLT_STR, &pctx->c_state_len);
OCIBNDPL(pctx->curp1, pctx->c_zip_bp[1], errhp,":c_zip",payP->c_zip,SIZ(payP->c_zip),
      SQLT_STR, &pctx->c_zip_len);
OCIBNDPL(pctx->curp1, pctx->c_phone_bp[1], errhp,":c_phone",payP->c_phone,
      SIZ(payP->c_phone), SQLT_STR, &pctx->c_phone_len);
OCIBNDPL(pctx->curp1, pctx->c_since_bp[1], errhp,":c_since",&payP->c_since,
      SIZ(OCIDate), SQLT_ODT, &pctx->c_since_len);
OCIBNDPL(pctx->curp1, pctx->c_credit_bp[1], errhp,":c_credit",payP->c_credit,
      SIZ(payP->c_credit),SQLT_CHR, &pctx->c_credit_len);
OCIBNDPL(pctx->curp1, pctx->c_credit_lim_bp[1],errhp,":c_credit_lim",
      ADR(payP->c_credit_lim),SIZ(int),SQLT_INT, &pctx->c_credit_lim_len);
OCIBNDPL(pctx->curp1, pctx->c_discount_bp[1], errhp,":c_discount",
      ADR(payP->c_discount),SIZ(double), SQLT_FLT, &pctx->c_discount_len);
OCIBNDPL(pctx->curp1, pctx->c_balance_bp[1], errhp,":c_balance",
      ADR(payP->c_balance), SIZ(payP->c_balance),SQLT_FLT, &pctx->c_balance_len);
OCIBNDPL(pctx->curp1, pctx->c_data_bp[1], errhp,":c_data",payP->c_data,SIZ(payP->c_data),
      SQLT_STR, &pctx->c_data_len);
/*
OCIBNDR(pctx->curp1, pctx->h_date_bp1, errhp,":h_date",payP->h_date,SIZ(payP->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(payP->retries),
      SIZ(int), SQLT_INT, &pctx->retries_len);
OCIBNDPL(pctx->curp1, pctx->cr_date_bp[1], errhp,":cr_date",ADR(payP->cr_date),
      SIZ(OCIDate),SQLT_ODT, &pctx->cr_date_len);

 return (0);

}



tkvcp (ora_cn_data_t *ora_SlotDataP)

{

 /*  */
 payctx *pctx = ora_SlotDataP->pctx;
 global_payment_t *payP = ora_SlotDataP->payP;
 OCIEnv *tpcenv = ora_SlotDataP->tpcenv;
 OCIServer *tpcsrv = ora_SlotDataP->tpcsrv;
 OCIError *errhp = ora_SlotDataP->errhp;
 OCISvcCtx *tpcsvc = ora_SlotDataP->tpcsvc;
 OCISession *tpcusr = ora_SlotDataP->tpcusr;
 OCIStmt *curi = ora_SlotDataP->curi;

retry:

 pctx->w_id_len = SIZ(payP->w_id);
 pctx->d_id_len = SIZ(payP->d_id);
 pctx->c_w_id_len = 0;
 pctx->c_d_id_len = 0;
 pctx->c_id_len = 0;
 pctx->h_amount_len = SIZ(payP->h_amount);
 pctx->c_last_len = SIZ(payP->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(payP->retries);
 pctx->cr_date_len = 7;

 if(payP->bylastname){
  payP->execstatus=OCIStmtExecute(tpcsvc,pctx->curp1,errhp,1,0,
            NULLP(CONST OCISnapshot),NULLP(OCISnapshot),
            OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS);
 } else {
  payP->execstatus=OCIStmtExecute(tpcsvc,pctx->curp0,errhp,1,0,
            NULLP(CONST OCISnapshot),NULLP(OCISnapshot),
            OCI_DEFAULT|OCI_COMMIT_ON_SUCCESS);
 }

 if(payP->execstatus!= OCI_SUCCESS) {
   OCITransRollback(tpcsvc,errhp,OCI_DEFAULT);
   payP->errcode = OCIERROR(errhp,payP->execstatus);
   if(payP->errcode == NOT_SERIALIZABLE) {
     payP->retries++;
     goto retry;
   } else if (payP->errcode == RECOVERR) {
     payP->retries++;
     goto retry;
   } else if (payP->errcode == SNAPSHOT_TOO_OLD) {
     payP->retries++;
     goto retry;
   } else {
     return -1;
   }
  }
 return 0;
}



void tkvcpdone (ora_cn_data_t *ora_SlotDataP)

{
 if(ora_SlotDataP->pctx) {
  free(ora_SlotDataP->pctx);
  ora_SlotDataP->pctx = NULL;
 }
 if (ora_SlotDataP->payP) {
  free(ora_SlotDataP->payP);
```

```
  ora_SlotDataP->payP = NULL;
 }
}
```

plsto.c

```
 #ifdef RCSID
static char *RCSid =
 "$Header: plsto.c 7010000.3 95/02/14 12:48:03 plai Generic<base> $ Copyr (c) 1994 Oracle for Encina";
#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 "tpcc.h"
#include "plora.h"
#include "tpccflags.h"

#ifdef PLSQLSTO
#define SQLTXT "BEGIN stocklevel.getstocklevel (:w_id, :d_id, :threshold, \
 :low_stock); END;"
#else
#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"
/* query using functional index*/
/*
#define SQLTXT "SELECT count (DISTINCT s_i_id) \
         FROM ordl, stok, dist \
         WHERE d_id = :d_id AND d_w_id = :w_id AND \
              d_id = ol_d_id AND d_w_id = ol_w_id AND \
              ol_o_id BETWEEN (d_next_o_id - 20) AND (d_next_o_id - 1) AND \
              decode(SIGN(s_quantity -21) , -1, s_w_id*100000 + s_i_id, NULL) \
                                  =   ol_w_id*100000 + ol_i_id  AND \
              s_quantity < :threshold;"

*/

#endif

struct stoctx {
 OCIStmt *curs;
 OCIBind *w_id_bp;
 OCIBind *d_id_bp;
 OCIBind *threshold_bp;
#ifdef PLSQLSTO
 OCIBind *low_stock_bp;
#else
 OCIDefine *low_stock_bp;
#endif
 int norow;
};

typedef struct stoctx stoctx;

/* stoctx *sctx; */


tkvcsinit (ora_cn_data_t *ora_SlotDataP)

{

 /*  */
 stoctx *sctx;
 global_stock_t *stoP;
 OCIEnv *tpcenv = ora_SlotDataP->tpcenv;
 OCIServer *tpcsrv = ora_SlotDataP->tpcsrv;
 OCIError *errhp = ora_SlotDataP->errhp;
 OCISvcCtx *tpcsvc = ora_SlotDataP->tpcsvc;
 OCISession *tpcusr = ora_SlotDataP->tpcusr;
 OCIStmt *curi = ora_SlotDataP->curi;

 text stmbuf[SQL_BUF_SIZE];
 sctx = (stoctx *)malloc(sizeof(stoctx));
 memset(sctx,(char)0,sizeof(stoctx));
 ora_SlotDataP->sctx = (void *)sctx;

 ora_SlotDataP->stoP = (global_stock_t *)malloc(sizeof(global_stock_t));
 memset(ora_SlotDataP->stoP,(char)0,sizeof(global_stock_t));
 stoP = ora_SlotDataP->stoP;

 sctx->norow=0;

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

  /* bind variables*/

  OCIBND(sctx->curs,sctx->w_id_bp,errhp, ":w_id", ADR(stoP->w_id),sizeof(int),
         SQLT_INT);
  OCIBND(sctx->curs,sctx->d_id_bp,errhp, ":d_id", ADR(stoP->d_id),sizeof(int),
    SQLT_INT);
  OCIBND(sctx->curs,sctx->threshold_bp,errhp, ":threshold", ADR(stoP->threshold),
    sizeof(int),SQLT_INT);
#ifdef PLSQLSTO
  OCIBND(sctx->curs,sctx->low_stock_bp,errhp,":low_stock" , ADR(stoP->low_stock),
    sizeof(int), SQLT_INT);
#else
  OCIDEFINE(sctx->curs,sctx->low_stock_bp,errhp, 1, ADR(stoP->low_stock),
    sizeof(int), SQLT_INT);
#endif

  return (0);

}



tkvcs (ora_cn_data_t *ora_SlotDataP)

{
 stoctx *sctx = (stoctx *)ora_SlotDataP->sctx;
 global_stock_t *stoP = ora_SlotDataP->stoP;
 OCIEnv *tpcenv = ora_SlotDataP->tpcenv;
 OCIServer *tpcsrv = ora_SlotDataP->tpcsrv;
 OCIError *errhp = ora_SlotDataP->errhp;
 OCISvcCtx *tpcsvc = ora_SlotDataP->tpcsvc;
 OCISession *tpcusr = ora_SlotDataP->tpcusr;
 OCIStmt *curi = ora_SlotDataP->curi;

retry:
   stoP->execstatus= OCIStmtExecute(tpcsvc,sctx->curs,errhp,1,0,0,0,
                        OCI_COMMIT_ON_SUCCESS | OCI_DEFAULT);
   if (stoP->execstatus != OCI_SUCCESS)
   {
     stoP->errcode=OCIERROR(errhp,stoP->execstatus);
     OCITransCommit(tpcsvc,errhp,OCI_DEFAULT);
     if((stoP->errcode == NOT_SERIALIZABLE) || (stoP->errcode == RECOVERR)
        || (stoP->errcode == SNAPSHOT_TOO_OLD))
     {
       stoP->retries++;
       goto retry;
     } else {
       return -1;
     }
   }

 return (0);
}



void tkvcsdone (ora_cn_data_t *ora_SlotDataP)

{
 /*  */
 stoctx *sctx = (stoctx *)ora_SlotDataP->sctx;
 if (sctx) {
   free(sctx);
   ora_SlotDataP->sctx = NULL;
 }
 if (ora_SlotDataP->stoP) {
   free(ora_SlotDataP->stoP);
   ora_SlotDataP->stoP = NULL;
 }

}
```

tkvcinin.sql

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

CREATE OR REPLACE PACKAGE inittpcc
AS
TYPE intarray IS TABLE OF INTEGER INDEX BY BINARY_INTEGER;
TYPE distarray IS TABLE OF VARCHAR(24) INDEX BY BINARY_INTEGER;
nulldate      DATE;
TYPE rowidarray IS TABLE OF ROWID INDEX BY PLS_INTEGER;
s_dist                                  distarray;
idx1arr                         intarray;
s_remote                        intarray;
```

```
dist                intarray;
row_id              rowidarray;
cust_rowid          rowid;
dist_name           VARCHAR2(11);
ware_name           VARCHAR2(11);
c_num               PLS_INTEGER;

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

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

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

 PROCEDURE init_pay IS
 BEGIN
   NULL;
 END init_pay;

END inittpcc;
/
show errors
exit
```

tkvcpnew.sql

```
-- New Order Anonymous block

 DECLARE
    idx                 PLS_INTEGER;
    dummy_local         PLS_INTEGER;
    cache_ol_cnt        PLS_INTEGER;
    not_serializable    EXCEPTION;
    PRAGMA EXCEPTION_INIT(not_serializable,-8177);
    deadlock            EXCEPTION;
    PRAGMA EXCEPTION_INIT(deadlock,-60);
    snapshot_too_old    EXCEPTION;
    PRAGMA EXCEPTION_INIT(snapshot_too_old,-1555);

 PROCEDURE u1 IS
 BEGIN
     FORALL idx IN 1 .. cache_ol_cnt
       UPDATE /*+ VECTOR_READ */ 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 /*+ VECTOR_READ */ stock_item
       SET s_order_cnt = s_order_cnt + 1,
       s_ytd = s_ytd + :ol_quantity(idx),
       s_remote_cnt = s_remote_cnt + :s_remote(idx),
       s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                     THEN  s_quantity +91
                     ELSE  s_quantity
                END) -  :ol_quantity(idx)
       WHERE i_id = :ol_i_id(idx)
       AND s_w_id = :ol_supply_w_id(idx)
       RETURNING i_price, i_name, s_quantity, s_dist_02,
             i_price*:ol_quantity(idx),
        CASE WHEN i_data NOT LIKE '%original%'
            THEN 'G'
             ELSE (CASE WHEN  s_data NOT LIKE '%original%'
                 THEN 'G'
                 ELSE 'B'
                 END)
        END
     BULK COLLECT INTO :i_price, :i_name, :s_quantity, inittpcc.s_dist,
                  :ol_amount,:brand_generic;
 END u2;

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

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

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

 PROCEDURE u6 IS
 BEGIN
     FORALL idx IN 1 .. cache_ol_cnt
       UPDATE /*+ VECTOR_READ */ stock_item
       SET s_order_cnt = s_order_cnt + 1,
       s_ytd = s_ytd + :ol_quantity(idx),
       s_remote_cnt = s_remote_cnt + :s_remote(idx),
       s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                     THEN  s_quantity +91
```

```
                    ELSE  s_quantity
                END) -  :ol_quantity(idx)
    WHERE i_id = :ol_i_id(idx)
    AND s_w_id = :ol_supply_w_id(idx)
    RETURNING i_price, i_name, s_quantity, s_dist_06,
            i_price*:ol_quantity(idx),
     CASE WHEN i_data NOT LIKE '%original%'
          THEN 'G'
          ELSE (CASE WHEN  s_data NOT LIKE '%original%'
               THEN 'G'
               ELSE 'B'
               END)
     END
   BULK COLLECT INTO :i_price, :i_name, :s_quantity, inittpcc.s_dist,
                :ol_amount,:brand_generic;
END u6;

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

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

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

PROCEDURE u10 IS
BEGIN
    FORALL idx IN 1 .. cache_ol_cnt
     UPDATE /*+ VECTOR_READ */ stock_item
     SET s_order_cnt = s_order_cnt + 1,
     s_ytd = s_ytd + :ol_quantity(idx),
     s_remote_cnt = s_remote_cnt + :s_remote(idx),
     s_quantity = (CASE WHEN s_quantity < :ol_quantity (idx) + 10
                    THEN  s_quantity +91
                    ELSE  s_quantity
                END) -  :ol_quantity(idx)
     WHERE i_id = :ol_i_id(idx)
     AND s_w_id = :ol_supply_w_id(idx)
     RETURNING i_price, i_name, s_quantity, s_dist_10,
             i_price*:ol_quantity(idx),
      CASE WHEN i_data NOT LIKE '%original%'
           THEN 'G'
           ELSE (CASE WHEN  s_data NOT LIKE '%original%'
                THEN 'G'
                ELSE 'B'
                END)
      END
    BULK COLLECT INTO :i_price, :i_name, :s_quantity, inittpcc.s_dist,
                :ol_amount,:brand_generic;
END u10;

PROCEDURE fix_items IS
  rows_lost                PLS_INTEGER;
  max_index                PLS_INTEGER;
  temp_index               PLS_INTEGER;
BEGIN
  idx := 1;
  rows_lost := 0;
  max_index := dummy_local;

  WHILE (max_index != cache_ol_cnt) LOOP

   WHILE (idx <= sql%rowcount AND
             sql%bulk_rowcount(idx + rows_lost) = 1)
   LOOP
    idx := idx + 1;
   END LOOP;

   temp_index := max_index;
   WHILE (temp_index >= idx + rows_lost) LOOP
     :ol_amount(temp_index + 1)     :=  :ol_amount(temp_index);
     :i_price(temp_index + 1)       :=  :i_price(temp_index);
     :i_name(temp_index + 1)        :=  :i_name(temp_index);
     :s_quantity(temp_index + 1)    :=  :s_quantity(temp_index);
     inittpcc.s_dist(temp_index + 1) :=  inittpcc.s_dist(temp_index);
     :brand_generic(temp_index + 1)  :=  :brand_generic(temp_index);
     temp_index := temp_index - 1;
   END LOOP;

   IF (idx + rows_lost <= cache_ol_cnt) THEN
     :i_price(idx + rows_lost)       :=  0;
     :i_name(idx + rows_lost)        :=  'NO ITEM';
     :s_quantity(idx + rows_lost)    :=  0;
     inittpcc.s_dist(idx + rows_lost) := NULL;
     :brand_generic(idx + rows_lost)  := ' ';
     :ol_amount(idx + rows_lost)     := 0;
     rows_lost := rows_lost + 1;
     max_index := max_index + 1;
   END IF;

 END LOOP;
END fix_items;

BEGIN
   LOOP BEGIN
     cache_ol_cnt := :o_ol_cnt;

     UPDATE dist SET d_next_o_id = d_next_o_id + 1
      WHERE d_id = :d_id AND  d_w_id = :w_id
      RETURNING d_tax, d_next_o_id-1
      INTO :d_tax, :o_id;

     SELECT c_discount, c_last, c_credit, w_tax
      INTO :c_discount, :c_last, :c_credit , :w_tax
      FROM cust , ware
      WHERE c_id = :c_id AND c_d_id = :d_id AND c_w_id = w_id
      AND  w_id = :w_id;


     INSERT INTO nord (no_o_id, no_d_id, no_w_id)
      VALUES (:o_id, :d_id, :w_id);

     INSERT INTO ordr  (o_id,o_d_id, o_w_id, o_c_id, o_entry_d,
                  o_carrier_id, o_ol_cnt, o_all_local)
      VALUES (:o_id, :d_id, :w_id, :c_id,
            :cr_date, 11, :o_ol_cnt, :o_all_local);


     dummy_local:=  :d_id;

     IF (dummy_local< 6) THEN
       IF (dummy_local< 3) THEN
         IF (dummy_local= 1) THEN
           u1;
         ELSE
           u2;
         END IF;
       ELSE
```

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

    dummy_local:= sql%rowcount;

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

    FORALL idx IN 1..dummy_local
     INSERT INTO ordl
       (ol_o_id, ol_d_id, ol_w_id, ol_number, ol_delivery_d,ol_i_id,
          ol_supply_w_id,ol_quantity,ol_amount,ol_dist_info)
     VALUES (:o_id, :d_id, :w_id, inittpcc.idx1arr(idx), inittpcc.nulldate,
          :ol_i_id(idx), :ol_supply_w_id(idx),
          :ol_quantity(idx), :ol_amount(idx), inittpcc.s_dist(idx));

    IF (dummy_local!= :o_ol_cnt) THEN
      :o_ol_cnt := dummy_local;
      ROLLBACK;
    END IF;

 EXIT;

 EXCEPTION
      WHEN not_serializableOR deadlock OR snapshot_too_old THEN
        ROLLBACK;
        :retry := :retry + 1;
    END;
  END LOOP;
END;
```

tpcc.h

```
/*
* $Header: tpcc.h 7030100.1 95/07/19 15:10:55 plai Generic<base> $ Copyr (c) 1993 Oracle for Encina
*/
/*+=================================================================+
|       Copyright (c) 1995  Oracle Corp, Redwood Shores, CA      |
|              OPEN SYSTEMS PERFORMANCE GROUP                    |
|                 All Rights Reserved                            |
+=================================================================+
| FILENAME
|    tpcc.h
| DESCRIPTION
|    Include file for TPC-C benchmark programs.
+=================================================================*/

#ifndef TPCC_H
#define TPCC_H

#ifndef FALSE
# define FALSE 0
#endif

#ifndef TRUE
# define TRUE 1
#endif

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

#ifndef boolean
#define boolean int
#endif

#include <oratypes.h>
#include <oci.h>
#include <ocidfn.h>
/*
#ifdef __STDC__
#include "ociapr.h"
#else
#include "ocikpr.h"
#endif
*/

typedef struct cda_def csrdef;
typedef struct cda_def ldadef;


/* TPC-C transaction functions */

extern int TPCinit ();
extern int TPCnew ();
extern int TPCpay ();
extern int TPCord ();
extern int TPCdel ();
extern int TPCsto ();
extern void TPCexit ();
extern int TPCdumpinit ();
extern void TPCdumpnew ();
extern void TPCdumppay ();
extern void TPCdumpord ();
extern void TPCdumpdel ();
extern void TPCdumpsto ();
extern void TPCdumpexit ();
/*extern void userlog(char* fmtp, ...); */


/* Error codes */

#define RECOVERR -10
#define IRRECERR -20
#define NOERR    111
#define DEL_ERROR -666
#define DEL_DATE_LEN 7
#define NDISTS 10
#define NITEMS 15
#define SQL_BUF_SIZE 8192

#define FULLDATE "dd-mon-yy.hh24:mi:ss"
#define SHORTDATE "dd-mm-yyyy"



#define DELRT 80.0

extern int tkvcninit ();
extern int tkvcpinit ();
extern int tkvcoinit ();
extern int tkvcdinit ();
extern int tkvcsinit ();

extern int tkvcn ();
extern int tkvcp ();
extern int tkvco ();
extern int tkvcd ();
extern int tkvcs ();

extern void tkvcndone ();
extern void tkvcpdone ();
extern void tkvcodone ();
extern void tkvcddone ();
extern void tkvcsdone ();

extern int tkvcss (); /* for alter session to get memory size and trace */
extern boolean multitranx;
extern int ord_init;


extern void errrpt ();
extern int ocierror(char *fname, int lineno,OCIError *errhp, sword status);
extern int sqlfile(char*fname, text *linebuf);

extern FILE *lfp;
extern FILE *fopen ();
extern int proc_no;
extern int doid[];
#if 0
extern int execstatus;
extern int errcode;

extern OCIEnv *tpcenv;
extern OCIServer *tpcsrv;
extern OCIError *errhp;
extern OCISvcCtx *tpcsvc;
extern OCISession *tpcusr;
extern OCIStmt *curntest;
/* The bind and define handles for each transaction are
  included in their respective header files.*/



/* for stock-level transaction */

extern int w_id;
extern int d_id;
extern int c_id;
extern int threshold;
extern int low_stock;
```

```
/* for delivery transaction */

extern int del_o_id[10];
extern int carrier_id;
extern int retries;

/* for order-status transaction */

extern int bylastname;
extern char c_last[17];
extern char c_first[17];
extern char c_middle[3];
extern double c_balance;
extern int o_id;
extern text o_entry_d[20];
extern int o_carrier_id;
extern int o_ol_cnt;
extern int ol_supply_w_id[15];
extern int ol_i_id[15];
extern int ol_quantity[15];
extern int ol_amount[15];
ub4    ol_del_len[15];
extern text ol_delivery_d[15][11];
/* xnie - begin */
extern OCIRowid *o_rowid;
/* xnie - end */

/* for payment transaction */

extern int c_w_id;
extern int c_d_id;
extern int h_amount;
extern char w_street_1[21];
extern char w_street_2[21];
extern char w_city[21];
extern char w_state[3];
extern char w_zip[10];
extern char d_street_1[21];
extern char d_street_2[21];
extern char d_city[21];
extern char d_state[3];
extern char d_zip[10];
extern char c_street_1[21];
extern char c_street_2[21];
extern char c_city[21];
extern char c_state[3];
extern char c_zip[10];
extern char c_phone[17];
extern text c_since_d[11];
extern char c_credit[3];
extern int c_credit_lim;
extern float c_discount;
extern char c_data[201];
extern text h_date[20];

/* for new order transaction */

extern int nol_i_id[15];
extern int nol_supply_w_id[15];
extern int nol_quantity[15];
extern int nol_quanti10[15];
extern int nol_quanti91[15];
extern int nol_ytdqty[15];
extern int nol_amount[15];
extern int o_all_local;
extern float w_tax;
extern float d_tax;
extern float total_amount;
extern char i_name[15][25];
extern int i_name_strlen[15];
extern ub2 i_name_strlen_len[15];
extern ub2 i_name_strlen_rcode[15];
extern ub4 i_name_strlen_csize;
extern int s_quantity[15];
extern char brand_gen[15];
extern ub2 brand_gen_len[15];
extern ub2 brand_gen_rcode[15];
extern ub4 brand_gen_csize;
extern int i_price[15];
extern char brand_generic[15][1];
extern int status;
extern int tracelevel;

/* Miscellaneous */
extern OCIDate cr_date;
extern OCIDate c_since;
extern OCIDate o_entry_d_base;
extern OCIDate ol_d_base[15];
#endif
#ifndef DISCARD
# define DISCARD (void)
#endif

#ifndef sword
# define sword int
#endif

#define VER7          2
#define NA            -1      /* ANSI SQL NULL */
#define NLT           1       /* length for string null terminator*/
#define DEADLOCK      60      /* ORA-00060: deadlock */
#define NO_DATA_FOUND  1403   /* ORA-01403: no data found */
#define NOT_SERIALIZABLE  8177  /* ORA-08177: transaction not serializable*/
#define SNAPSHOT_TOO_OLD  1555  /* ORA-01555: snapshot too old */

#ifndef NULLP
# define NULLP(x)  (x  *)NULL
#endif /* NULLP */

#define ADR(object) ((ub1 *)&(object))
#define SIZ(object) ((sword)sizeof(object))

typedef char date[24+NLT];
typedef char varchar2;

#define min(x,y)  (((x) < (y)) ? (x) : (y))

#define OCIERROR(errp,function)\
      ocierror(__FILE__,__LINE__,(errp),(function));

#define OCIBND(stmp, bndp, errp, sqlvar, progv, progvl, ftype)\
    ocierror(__FILE__,__LINE__,(errp), \
        OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
    ocierror(__FILE__,__LINE__, (errp), \
        OCIBindByName((stmp), &(bndp), (errp), \
                                    (text *)(sqlvar), strlen((sqlvar)),\
                                    (progv), (progvl), (ftype),0,0,0,0,0,OCI_DEFAULT));



/* bind arrays for sql */
#define OCIBNDRA(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode) \
    DISCARD ocierror(__FILE__,__LINE__,(errp), \
     OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
    DISCARD ocierror(__FILE__,__LINE__,(errp), \
      OCIBindByName((stmp),&(bndp),(errp),(text*)(sqlvar),strlen((sqlvar)),\

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


/* use with callback data */
#define OCIBNDRAD(stmp,bndp,errp,sqlvar,progvl,ftype,indp,ctxp,\
            cbf_nodata,cbf_data) \
    DISCARD ocierror(__FILE__,__LINE__,(errp), \
     OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
    DISCARD ocierror(__FILE__,__LINE__,(errp), \
       OCIBindByName((stmp),&(bndp),(errp),(text*)(sqlvar), \
                strlen((sqlvar)),0,(progvl),(ftype), \
                indp,0,0,0,0,OCI_DATA_AT_EXEC)); \
    DISCARD ocierror(__FILE__,__LINE__,(errp), \
      OCIBindDynamic((bndp),(errp),(ctxp),(cbf_nodata),(ctxp),(cbf_data)));



/* bind in/out for plsql without indicator and rcode */
#define OCIBNDPL(stmp,bndp,errp,sqlvar,progv,progvl,ftype,alen)\
  DISCARD ocierror(__FILE__,__LINE__,(errp), \
    OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
  DISCARD ocierror(__FILE__,__LINE__,(errp), \
    OCIBindByName((stmp),&(bndp),(errp),(CONST text *)(sqlvar), \
     (sb4)strlen((CONST char *)(sqlvar)), (dvoid*)(progv),(progvl),(ftype),\
         NULLP(dvoid),(alen), NULLP(ub2), 0,NULLP(ub4),OCI_DEFAULT));

/* bind  in values for plsql with indicator and rcode */
#define OCIBNDR(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode) \
    DISCARD ocierror(__FILE__,__LINE__,(errp), \
     OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0)); \
    DISCARD ocierror(__FILE__,__LINE__,(errp), \
     OCIBindByName((stmp),&(bndp),(errp),(text*)(sqlvar),strlen((sqlvar)),\
                                    (progv),(progvl),(ftype),(indp),(alen),(arcode),0,0, \
            OCI_DEFAULT));

/* bind in/out for plsql arrays witout indicator and rcode */
#define OCIBNDPLA(stmp,bndp,errp,sqlvar,progv,progvl,ftype,alen,ms,cu)\
    DISCARD ocierror(__FILE__,__LINE__, (errp), \
      OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0));\
    DISCARD ocierror(__FILE__,__LINE__,(errp),\
      OCIBindByName((stmp),&(bndp),(errp),(CONST text *)(sqlvar), \
          (sb4)strlen((CONST char *) (sqlvar)),(void *)(progv), \
          (progvl),(ftype),NULL,(alen),NULL,(ms),(cu),OCI_DEFAULT));

/* bind in/out values for plsql with indicator and rcode */
#define OCIBNDRAA(stmp,bndp,errp,sqlvar,progv,progvl,ftype,indp,alen,arcode,\
            ms,cu) \
    ocierror(__FILE__,__LINE__, (errp), \
      OCIHandleAlloc((stmp),(dvoid**)&(bndp),OCI_HTYPE_BIND,0,(dvoid**)0));\
    ocierror(__FILE__,__LINE__,(errp),\
      OCIBindByName((stmp),&(bndp),(errp),(text*)(sqlvar),strlen((sqlvar)),\
    (progv),(progvl),(ftype),(indp),(alen),(arcode),(ms),(cu),OCI_DEFAULT));

#define OCIDEFINE(stmp,dfnp,errp,pos,progv,progvl,ftype)\
    OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),(progvl),(ftype),\
                                    0,0,0,OCI_DEFAULT);


#define OCIDEF(stmp,dfnp,errp,pos,progv,progvl,ftype) \
    OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTYPE_DEFINE,0,\
```

```
                                                    (dvoid**)0);\
    OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),(progvl),\
                   (ftype),NULL,NULL,NULL,OCI_DEFAULT);\


#define OCIDFNRA(stmp,dfnp,errp,pos,progv,progvl,ftype,indp,alen,arcode) \
    OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTYPE_DEFINE,0,\
                                                    (dvoid**)0);\
    OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv),\
                                                    (progvl),(ftype),(indp),(alen),\
                                                    (arcode),OCI_DEFAULT);

#define OCIDFNDYN(stmp,dfnp,errp,pos,progv,progvl,ftype,indp,ctxp,cbf_data) \
    ocierror(__FILE__,__LINE__,(errp), \
    OCIHandleAlloc((stmp),(dvoid**)&(dfnp),OCI_HTYPE_DEFINE,0,\
                                                    (dvoid**)0));\
    ocierror(__FILE__,__LINE__,(errp), \
    OCIDefineByPos((stmp),&(dfnp),(errp),(pos),(progv), (progvl),(ftype),\
                        (indp),NULL,NULL, OCI_DYNAMIC_FETCH));\
    ocierror(__FILE__,__LINE__,(errp), \
    OCIDefineDynamic((dfnp),(errp),(ctxp),(cbf_data)));


#if 0

/* New order */

struct newinstruct {
  int w_id;
  int d_id;
  int c_id;
  int ol_i_id[15];
  int ol_supply_w_id[15];
  int ol_quantity[15];
};

struct newoutstruct {
  int terror;
  int o_id;
  int o_ol_cnt;
  char c_last[17];
  char c_credit[3];
  float c_discount;
  float w_tax;
  float d_tax;
  char o_entry_d[20];
  float total_amount;
  char i_name[15][25];
  int s_quantity[15];
  char brand_generic[15];
  float i_price[15];
  float ol_amount[15];
  char status[26];
  int retry;
};

struct newstruct {
  struct newinstruct newin;
  struct newoutstruct newout;
};


/* Payment */

struct payinstruct {
  int w_id;
  int d_id;
  int c_w_id;
  int c_d_id;
  int c_id;
  int bylastname;
  int h_amount;
  char c_last[17];
};

struct payoutstruct {
  int terror;
  char w_street_1[21];
  char w_street_2[21];
  char w_city[21];
  char w_state[3];
  char w_zip[10];
  char d_street_1[21];
  char d_street_2[21];
  char d_city[21];
  char d_state[3];
  char d_zip[10];
  int  c_id;
  char c_first[17];
  char c_middle[3];
  char c_last[17];
  char c_street_1[21];
  char c_street_2[21];
  char c_city[21];
  char c_state[3];
  char c_zip[10];
  char c_phone[17];
  char c_since[11];
  char c_credit[3];
  double c_credit_lim;
  float  c_discount;
  double c_balance;
  char c_data[201];
  char h_date[20];
  int retry;
};

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


/* Order status */

struct ordinstruct {
  int w_id;
  int d_id;
  int c_id;
  int bylastname;
  char c_last[17];
};

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

struct ordstruct {
  struct ordinstruct ordin;
  struct ordoutstruct ordout;
};


/* Delivery */

struct delinstruct {
  int w_id;
  int o_carrier_id;
  double qtime;
  int in_timing_int;
};

struct deloutstruct {
  int terror;
  int retry;
};

struct delstruct {
  struct delinstruct delin;
  struct deloutstruct delout;
};


/* Stock level */

struct stoinstruct {
  int w_id;
  int d_id;
  int threshold;
};

struct stooutstruct {
  int terror;
  int low_stock;
  int retry;
};

struct stostruct {
  struct stoinstruct stoin;
  struct stooutstruct stoout;
};
#endif
#endif
```

tpcc_info.h

```
  /*
* $Header: tpcc_info.h 7030100.1 95/07/19 15:11:37 plai Generic<base> $ Copyr (c) 1995 Oracle for
Encina
```

```
*/
/*+=================================================================+
|        Copyright (c) 1995  Oracle Corp, Redwood Shores, CA      |
|                OPEN SYSTEMS PERFORMANCE GROUP                   |
|                   All Rights Reserved                           |
+=================================================================+
| FILENAME
|    tpcc_info.h
| DESCRIPTION
|    Include file for TPC-C benchmark programs.
+=================================================================*/

#ifndef TPCC_INFO_H
#define TPCC_INFO_H

/* this set is duplicated in c_Defs.h, c_Defs.h is used for batch driver */
#define MENTXN    0          /* menu txn */
#define NEWTXN    1          /* new order transaction */
#define PAYTXN    2          /* payment transaction */
#define ORDTXN    3          /* order status transaction */
#define DELTXN    4          /* delivery transaction */
#define STOTXN    5          /* stock level transaction */
#define ALLTXN    6          /* for processing all txns */
#define ALLTXNNODEL 7        /* for processing all txns except delivery */
/* New order */

struct newinstruct {
  int w_id;
  int d_id;
  int c_id;
  int ol_i_id[15];
  int ol_supply_w_id[15];
  int ol_quantity[15];
};

struct newoutstruct {
  int terror;
  int o_id;
  int o_ol_cnt;
  char c_last[17];
  char c_credit[3];
  float c_discount;
  float w_tax;
  float d_tax;
  char o_entry_d[20];
  float total_amount;
  char i_name[15][25];
  int s_quantity[15];
  char brand_generic[15];
  float i_price[15];
  float ol_amount[15];
  char status[26];
  int retry;
};

struct newstruct {
  int    retval;
  int old_quantity[15];
  struct newinstruct newin;
  struct newoutstruct newout;
};


/* Payment */

struct payinstruct {
  int w_id;
  int d_id;
  int c_w_id;
  int c_d_id;
  int c_id;
  int bylastname;
  int h_amount;
  char c_last[17];
};

struct payoutstruct {
  int terror;
  char w_street_1[21];
  char w_street_2[21];
  char w_city[21];
  char w_state[3];
  char w_zip[10];
  char d_street_1[21];
  char d_street_2[21];
  char d_city[21];
  char d_state[3];
  char d_zip[10];
  int  c_id;
  char c_first[17];
  char c_middle[3];
  char c_last[17];
  char c_street_1[21];
  char c_street_2[21];
  char c_city[21];
  char c_state[3];
  char c_zip[10];
  char c_phone[17];
  char c_since[11];
  char c_credit[3];
  double c_credit_lim;
  float c_discount;
  double c_balance;
  char c_data[201];
  char h_date[20];
  int retry;
};

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


/* Order status */

struct ordinstruct {
  int w_id;
  int d_id;
  int c_id;
  int bylastname;
  char c_last[17];
};

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

struct ordstruct {
  int retval;
  struct ordinstruct ordin;
  struct ordoutstruct ordout;
};


/* Delivery */

struct delinstruct {
  int w_id;
  int o_carrier_id;
  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;
};

#endif
```

tpccflags.h

```
/*
* $Header: tpccflags.h for Encina
*/

#define PLSQLNO
#define DMLRETDEL
```

tpccpl.c

```
#ifdef RCSID
static char *RCSid =
  "$Header: /afs/transarc.com/project/encina/rcs/test/src/benchmarks/tpcc/server/ora10.1_mt/RCS/tpccpl.c,v
1.7 1999/05/26 16:29:59 wenjian Exp $ 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 <stdio.h>
#include <time.h>
#include "tpcc.h"
#ifdef TUX
#include <userlog.h>
#else
#include <stdarg.h>
#endif
#ifdef MULTI_THREADED
#include <dce/pthread.h>
#endif
#include "plora.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"
#define SQLTXTIMU "alter session set \"_in_memory_undo\" = true"
#define SQLTXTNCP "alter session set plsql_compiler_flags=anonymous_block_native"

int proc_no;
static char *db_uid;
static char *db_pwd;

#ifndef AVOID_DEADLOCK
void q_sort(global_newOrder_t *newP,struct newstruct *str,int left, int right, int *indx, int in);
void swap(global_newOrder_t *newP, struct newstruct *str, int i, int j, int *indx, int in);
#endif

/*
extern char oracle_home[256];
*/

/* NewOrder Binding stuff */

/** Delivery fileinfomation: Global.
 * One output file for deliveries for the server
 */
static char delivery_file_name[100];

#ifdef MULTI_THREADED
pthread_mutex_t dvry_log_lock;
#define DVRY_LOCK pthread_mutex_lock(&dvry_log_lock);
#define DVRY_UNLOCK pthread_mutex_unlock(&dvry_log_lock);
#define DVRY_LOCK_INIT pthread_mutex_init(&dvry_log_lock,pthread_mutexattr_default);
#else
#define DVRY_LOCK
#define DVRY_UNLOCK
#define DVRY_LOCK_INIT
#endif
FILE *lfp;
FILE *fopen ();

#ifdef ORA_NT
extern double dpbtimef();

#define gettime dpbtimef
#else
double gettime ();
#endif

/** Initialization of one connection */
static void initOCIhandles(ora_cn_data_t *cn_dataP, char* uid, char *pwd);

static int init_cn_data(ora_cn_data_t *dataP);

extern char oracle_home[256];

/* NewOrder Binding stuff */

#ifndef TUX
void userlog (char* fmtp, ...)
{
 va_list va;
 va_start(va,fmtp);
 vfprintf(stderr,fmtp,va);
 va_end(va);
}
#endif

/* vmm313 void ocierror(fname, lineno, errhp, status)  */
int ocierror(fname, lineno, errhp, status)
char *fname;
int lineno;
OCIError *errhp;
sword status;
{
 text errbuf[512];
 ub4 buflen;
 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);
/* vmm313    TPCexit(1); */
/* vmm313    exit(1); */
 case OCI_INVALID_HANDLE:
   fprintf(stderr,"Module %s Line %d\n", fname, lineno);
   fprintf(stderr,"Error - OCI_INVALID_HANDLE\n");
   TPCexit(1);
   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);
}


FILE  *vopen(fnam,mode)
char  *fnam;
char  *mode;
{
FILE *fd;

#ifdef  DEBUG
 fprintf(stderr, "tkvuopen() fnam: %s,   mode: %s\n", fnam, mode);
#endif

   fd = fopen((char *)fnam,(char *)mode);
   if (!fd){
       fprintf(stderr," fopen on %s failed %d\n",fnam,fd);
       exit(-1);
   }
   return(fd);
}

int sqlfile(fnam,linebuf)
char   *fnam;
```

```
text   *linebuf;
{
FILE *fd;
int nulpt = 0;
char realfile[512];

#ifdef DEBUG
 fprintf(stderr, "sqlfile() fnam: %s,  linebuf: %#x\n", fnam, linebuf);
#endif

  sprintf(realfile,"%s/bench/tpcc/benchrun/blocks/%s",oracle_home,fnam);

  fd = vopen(realfile,"r");
  while (fgets((char *)linebuf+nulpt, SQL_BUF_SIZE,fd))
  {
     nulpt = strlen((char *)linebuf);
  }
  return(nulpt);
}

#ifdef NOT
void vgetdate (unsigned char *oradt)
{
struct tm *loctime;
time_t  int_time;

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

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

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

/* Convert the current date and time into local time */
loctime = localtime( &int_time);

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

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


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

return;

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

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


      int day,month,year;

      memcpy(&Date,oradt,7);

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

      return;
}
void cvtdmyhms(unsigned char *oradt, char *outdate)
{

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


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

      memcpy(&Date,oradt,7);

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

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

      return;
}
#endif


void TPCexit (void)

{

  if (lfp) {
     fclose (lfp);
     lfp = NULL;
  }

}

/*  clean_connection
*
*  Called to clean a connection.
*  When using pthread this is registered during pthread_create
*  and called automatically bypthread when the thread exits.
*/
void clean_connection(void *ptr)
{

  /* free trans specific cursor handles first and later theora handles */
  ora_cn_data_t *cn_dataP = (ora_cn_data_t *)ptr;

  if (cn_dataP != NULL) {
      OCIServer *tpcsrv;
      OCISession *tpcusr;
      OCIEnv *tpcenv;
      OCIError *errhp;
      OCISvcCtx *tpcsvc;

      fprintf(stderr, "clean_connection, Freeing OCI handles\n");
      tkvcndone(cn_dataP);
      tkvcpdone(cn_dataP);
      tkvcodone(cn_dataP);
      tkvcddone(cn_dataP);
      tkvcsdone(cn_dataP);

      /* free OCI handles */
      if (tpcusr = cn_dataP->tpcusr) {
            fprintf(stderr, "free_handles> OCIHandleFree tpcusr\n");
            OCIHandleFree((dvoid *)tpcusr, OCI_HTYPE_SESSION);
      }
      if (tpcsvc = cn_dataP->tpcsvc) {
            fprintf(stderr, "free_handles> OCIHandleFree tpcsvc\n");
            OCIHandleFree((dvoid *)tpcsvc, OCI_HTYPE_SVCCTX);
      }
      if (errhp = cn_dataP->errhp) {
            fprintf(stderr, "free_handles> OCIHandleFree errhp\n");
            OCIHandleFree((dvoid *)errhp, OCI_HTYPE_ERROR);
      }
      if (tpcsrv = cn_dataP->tpcsrv) {
            fprintf(stderr, "free_handles> OCIHandleFree tpcsrv\n");
            OCIHandleFree((dvoid *)tpcsrv, OCI_HTYPE_SERVER);
      }
      if (tpcenv = cn_dataP->tpcenv) {
            fprintf(stderr, "free_handles> OCIHandleFree tpcenv\n");
            OCIHandleFree((dvoid *)tpcenv, OCI_HTYPE_ENV);
      }

      fprintf(stderr, "free_handles> free cn_dataP\n");
  }
}

TPCinit (id, uid, pwd, dvryFileName)
int id;
```

```
char *uid;
char *pwd;
char *dvryFileName;
{

  int i;
  text stmbuf[100];

  fprintf(stderr, "TPCinit id %d, uid %s pwd %s\n", id, uid, pwd);

  DVRY_LOCK_INIT;

  proc_no = id;
  db_uid = (char *)calloc(strlen(uid) + 1, sizeof(char));
  strcpy(db_uid, uid);
  db_pwd = (char *)calloc(strlen(pwd) + 1, sizeof(char));
  strcpy(db_pwd, pwd);

  err_printf("dvryFileNameis %s\n", dvryFileName);
  sprintf (delivery_file_name,"%s.%d", dvryFileName,proc_no);
  err_printf("delivery_file_nameis %s\n", delivery_file_name);
#ifdef USE_ORACLE_WAY
  if ((lfp = fopen (delivery_file_name,"w")) == NULL) {
#ifdef TUX
    userlog ("Error in TPC-C server %d: Failed to open %s\n",
            proc_no, delivery_file_name);
#else
    fprintf (stderr, "Error in TPC-C server %d: Failed to open %s\n",
            proc_no, delivery_file_name);
#endif
    return (-1);
  }
#endif

  OCIInitialize(OCI_DEFAULT|OCI_OBJECT,(dvoid*)0,0,0,0); /* check tpccpl.c */

  return (0);

}

static void initOCIhandles(ora_cn_data_t *cn_dataP, char *uid, char *pwd)
{

  int tracelevel = 0;  /* new define */
  OCIDate cr_date;
  text stmbuf[100];
  OCIEnv *tpcenv;
  OCIServer *tpcsrv;
  OCIError *errhp;
  OCISvcCtx *tpcsvc;
  OCISession *tpcusr;
  OCIStmt *curi;

  OCIEnvInit(&tpcenv, OCI_DEFAULT, 0, (dvoid **)0);
  OCIHandleAlloc((dvoid*)tpcenv, (dvoid **)&tpcsrv, OCI_HTYPE_SERVER, 0 , (dvoid **)0);
  OCIHandleAlloc((dvoid*)tpcenv, (dvoid **)&errhp, OCI_HTYPE_ERROR, 0 , (dvoid **)0);
  OCIHandleAlloc((dvoid*)tpcenv, (dvoid **)&tpcsvc, OCI_HTYPE_SVCCTX, 0 , (dvoid **)0);
  OCIServerAttach(tpcsrv, errhp, (text *)0,0,OCI_DEFAULT);
  OCIAttrSet((dvoid *)tpcsvc, OCI_HTYPE_SVCCTX, (dvoid *)tpcsrv, (ub4)0,OCI_ATTR_SERVER,
errhp);
  OCIHandleAlloc((dvoid*)tpcenv, (dvoid **)&tpcusr, OCI_HTYPE_SESSION, 0 , (dvoid **)0);
  OCIAttrSet((dvoid *)tpcusr, OCI_HTYPE_SESSION, (dvoid *)uid,
(ub4)strlen(uid),OCI_ATTR_USERNAME, errhp);
  OCIAttrSet((dvoid *)tpcusr, OCI_HTYPE_SESSION, (dvoid *)pwd, (ub4)strlen(pwd),
                OCI_ATTR_PASSWORD, errhp);
  OCIERROR(errhp, OCISessionBegin(tpcsvc, errhp, tpcusr, OCI_CRED_RDBMS, OCI_DEFAULT));

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

  /* run all transaction in serializable mode */

  OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);
  sprintf ((char *) stmbuf, SQLTXT);
  OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT);
  OCIERROR(errhp,OCIStmtExecute(tpcsvc, curi, errhp,1,0,0,0,OCI_DEFAULT));
  OCIHandleFree(curi, OCI_HTYPE_STMT);

  if (getenv("USE_IMU")) {
    printf("Use in_memory_undo\n");
    OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);
    sprintf ((char *) stmbuf, SQLTXTIMU);
    OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT);
    OCIERROR(errhp,OCIStmtExecute(tpcsvc, curi, errhp,1,0,0,0,OCI_DEFAULT));
    OCIHandleFree(curi, OCI_HTYPE_STMT);
  }

  if (getenv("USE_NCOMP")) {
    printf("Use NCOMP\n");
    OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);
    sprintf ((char *) stmbuf, SQLTXTNCP);
    OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf), OCI_NTV_SYNTAX, OCI_DEFAULT);
    OCIERROR(errhp,OCIStmtExecute(tpcsvc, curi, errhp,1,0,0,0,OCI_DEFAULT));
    OCIHandleFree(curi, OCI_HTYPE_STMT);
  }

/*
This is done in cvdrv.c
  if (tracelevel == 2) {
    OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);
    memset(stmbuf,0,100);
    sprintf ((char *) stmbuf, SQLTXTTRC);
    OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf),
                                        OCI_NTV_SYNTAX, OCI_DEFAULT);
    OCIERROR(errhp, OCIStmtExecute(tpcsvc, curi, errhp,1,0,0,0,OCI_DEFAULT));
    OCIHandleFree((dvoid *)curi, OCI_HTYPE_STMT);
  }
*/
  if (tracelevel == 3) {
    OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);
    memset(stmbuf,0,100);
    sprintf ((char *) stmbuf, SQLTXTTIM);
    OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf),
                                        OCI_NTV_SYNTAX, OCI_DEFAULT);
    OCIERROR(errhp, OCIStmtExecute(tpcsvc, curi, errhp,1,0,0,0,OCI_DEFAULT));
    OCIHandleFree((dvoid *)curi, OCI_HTYPE_STMT);
  }


  OCIERROR(errhp,OCIDateSysDate(errhp,&cr_date));

  /*  Store the handles just initialized in the thread slot
  */
  cn_dataP->tpcenv = tpcenv;
  cn_dataP->tpcsrv = tpcsrv;
  cn_dataP->errhp  = errhp;
  cn_dataP->tpcsvc = tpcsvc;
  cn_dataP->tpcusr = tpcusr;
  cn_dataP->curi   = curi;
}

/*
*  init_cn_data
*      Initializes all the transactions for a single connection
*/
static int init_cn_data(ora_cn_data_t *cnP)
{
  int status;

  initOCIhandles(cnP,db_uid, db_pwd);

  if (status = tkvcninit (cnP)) {  /* new order */
    fprintf(stderr, "tkvcninit failed: %d\n", status);
    TPCexit ();
    return (status);
  }

  if (status = tkvcpinit (cnP)) {   /* payment */
    fprintf(stderr, "tkvcpinit failed: %d\n", status);
    TPCexit ();
    return (status);
  }

  if (status = tkvcoinit (cnP)) {   /* order status */
    fprintf(stderr, "tkvcoinit failed: %d\n", status);
    TPCexit ();
    return (status);
  }

  if (status = tkvcdinit (cnP)) {   /* delivery */
    fprintf(stderr, "tkvcdinit failed: %d\n", status);
    TPCexit ();
    return (status);
  }

  if (status = tkvcsinit (cnP)) {   /* stock level */
    fprintf(stderr, "tkvcsinit failed: %d\n", status);
    TPCexit ();
    return (status);
  }
  return 0;
}

void *create_ora_connection() {
  ora_cn_data_t *cnP = (ora_cn_data_t *)malloc(sizeof(ora_cn_data_t));
  init_cn_data(cnP);
  return (void *)cnP;
}

/*****   The Transaction Code  *****/

TPCnew (cnP, str)
  void *cnP;
  struct newstruct *str;
{
  int i;
  int indx[NITEMS], ordl_cnt;
  ora_cn_data_t *cn_dataP = (ora_cn_data_t *)cnP;
  global_newOrder_t *newP = cn_dataP->globals;
  OCIError *errhp = cn_dataP->errhp;

  newP->w_id = str->newin.w_id;
  newP->d_id = str->newin.d_id;
  newP->c_id = str->newin.c_id;
  for (i = 0; i < 15; i++) {
    newP->nol_i_id[i] = str->newin.ol_i_id[i];
    newP->nol_supply_w_id[i] = str->newin.ol_supply_w_id[i];
    newP->nol_quantity[i] = str->newin.ol_quantity[i];
  }
```

```
 newP->retries = 0;

#ifndef AVOID_DEADLOCK

 for (i = NITEMS; i > 0; i--) {
   if (newP->nol_i_id[i-1] > 0) {
     ordl_cnt = i;
     break;
   }
 }

 for (i = 0; i < NITEMS; i++) indx[i] = i;

 q_sort(newP, str, 0, ordl_cnt-1, indx, 1);

#endif

/*
 vgetdate(newP->cr_date); */

 OCIERROR(errhp,OCIDateSysDate(errhp,&newP->cr_date));

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

 /* fill in date for o_entry_d from time in beginning of txn*/

/*
 cvtdmyhms(newP->cr_date,newP->o_entry_d);
*/
  newP->datelen = sizeof(newP->o_entry_d);
/*
  fprintf (stderr, "%s line %d: &newP->cr_date %ld (text*)FULLDATE %s SIZ(FULLDATE) %d
&newP->datelen %d newP->o_entry_d %s\n",__FILE__,__LINE__,
  &newP->cr_date,(char *)FULLDATE,strlen(FULLDATE),&newP->datelen,(char*)newP->o_entry_d);
*/
/*
sword OCIDateToText ( OCIError           *err,
                      CONST OCIDate      *date,
                      CONST text         *fmt,
                      ub1                fmt_length,
                      CONST text         *lang_name,
                      ub4                lang_length,
                      ub4                *buf_size,
                      text               *buf );


*/
  OCIERROR(errhp,
    OCIDateToText(errhp,&newP->cr_date,(text*)FULLDATE,SIZ(FULLDATE),(text*)0,0,
                  &newP->datelen,newP->o_entry_d));

 str->newout.terror = NOERR;
 str->newout.o_id = newP->o_id;
 str->newout.o_ol_cnt = newP->o_ol_cnt;
 strncpy (str->newout.c_last, newP->c_last, 17);
 strncpy (str->newout.c_credit, newP->c_credit, 3);
 str->newout.c_discount = newP->c_discount;
 str->newout.w_tax = (float)newP->w_tax;
 str->newout.d_tax = (float)newP->d_tax;
 strncpy (str->newout.o_entry_d, (char*)newP->o_entry_d, 20);
 str->newout.total_amount = newP->total_amount;
 for (i = 0; i < newP->o_ol_cnt; i++) {
   strncpy (str->newout.i_name[i], newP->i_name[i], 25);
   str->newout.s_quantity[i] = newP->s_quantity[i];
   str->newout.brand_generic[i] = newP->brand_gen[i];
   str->newout.i_price[i] = (float)(newP->i_price[i])/100;
   str->newout.ol_amount[i] = (float)(newP->nol_amount[i])/100;
 }

#ifndef AVOID_DEADLOCK
  q_sort(newP, str, 0, ordl_cnt-1, indx, 0);
#endif

 if (newP->status)
   strcpy (str->newout.status, "Item number is not valid");
 else
   str->newout.status[0] = '\0';
 str->newout.retry = newP->retries;
#if defined(TOP) || defined(TUX)   /* changed mjb 17 feb for tuxedo */
 return(1);
#else
 return (0);
#endif

}

TPCpay (cnP, str)
void *cnP;
struct paystruct *str;

{
 ora_cn_data_t *cn_dataP = (ora_cn_data_t *)cnP;
 global_payment_t *payP = cn_dataP->payP;
 OCIError *errhp = cn_dataP->errhp;
  payP->w_id = str->payin.w_id;
  payP->d_id = str->payin.d_id;
  payP->c_w_id = str->payin.c_w_id;
  payP->c_d_id = str->payin.c_d_id;
  payP->h_amount = str->payin.h_amount;
  payP->bylastname= str->payin.bylastname;

 /*
  vgetdate(payP->cr_date);  */
  OCIERROR(errhp,OCIDateSysDate(errhp,&payP->cr_date));

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

  if (str->payout.terror = tkvcp (cn_dataP)) {
    if (str->payout.terror != RECOVERR)
      str->payout.terror = IRRECERR;
    return (-1);
  }

 /*
  cvtdmyhms(cr_date,h_date);
 */
      payP->hlen=SIZ(payP->h_date);
      OCIERROR(errhp,OCIDateToText(errhp,&payP->cr_date,
           (text*)FULLDATE,strlen(FULLDATE),(text*)0,0,&payP->hlen,payP->h_date));

 /*
  cvtdmy(c_since,c_since_d);
 */
      payP->sincelen=SIZ(payP->c_since_d);
      OCIERROR(errhp,OCIDateToText(errhp,&payP->c_since,
        (text*)SHORTDATE,strlen(SHORTDATE),(text*)0,0,&payP->sincelen,payP->c_since_d));

  str->payout.terror = NOERR;
  strncpy (str->payout.w_street_1, payP->w_street_1, 21);
  strncpy (str->payout.w_street_2, payP->w_street_2, 21);
  strncpy (str->payout.w_city, payP->w_city, 21);
  strncpy (str->payout.w_state, payP->w_state, 3);
  strncpy (str->payout.w_zip, payP->w_zip, 10);
  strncpy (str->payout.d_street_1, payP->d_street_1, 21);
  strncpy (str->payout.d_street_2, payP->d_street_2, 21);
  strncpy (str->payout.d_city, payP->d_city, 21);
  strncpy (str->payout.d_state, payP->d_state, 3);
  strncpy (str->payout.d_zip, payP->d_zip, 10);
  str->payout.c_id = payP->c_id;
  strncpy (str->payout.c_first, payP->c_first, 17);
  strncpy (str->payout.c_middle, payP->c_middle, 3);
  strncpy (str->payout.c_last, payP->c_last, 17);
  strncpy (str->payout.c_street_1, payP->c_street_1, 21);
  strncpy (str->payout.c_street_2, payP->c_street_2, 21);
  strncpy (str->payout.c_city, payP->c_city, 21);
  strncpy (str->payout.c_state, payP->c_state, 3);
  strncpy (str->payout.c_zip, payP->c_zip, 10);
  strncpy (str->payout.c_phone, payP->c_phone, 17);
  strncpy (str->payout.c_since, (char*)payP->c_since_d, 11);
  strncpy (str->payout.c_credit, payP->c_credit, 3);
  str->payout.c_credit_lim = (float)(payP->c_credit_lim)/100;
  str->payout.c_discount = payP->c_discount;
  str->payout.c_balance = (float)(payP->c_balance)/100;
  strncpy (str->payout.c_data, payP->c_data, 201);
  strncpy (str->payout.h_date, (char*)payP->h_date, 20);
  str->payout.retry = payP->retries;
 #if defined(TOP) || defined(TUX)   /* changed mjb 17 Feb */
  return(1);
 #else
  return (0);
 #endif

 }

 TPCord (cnP, str)
 void *cnP;
 struct ordstruct *str;

 {
  ora_cn_data_t *cn_dataP = (ora_cn_data_t *)cnP;
  global_order_t *ordP = cn_dataP->ordP;
  int i;

  ordP->w_id = str->ordin.w_id;
  ordP->d_id = str->ordin.d_id;
  ordP->bylastname = str->ordin.bylastname;
  if (ordP->bylastname) {
    ordP->c_id = 0;
    strncpy (ordP->c_last, str->ordin.c_last, 17);
  }
  else {
    ordP->c_id = str->ordin.c_id;
    strcpy (ordP->c_last, " ");
  }
  ordP->retries = 0;
```

```
 if (str->ordout.terror = tkvco (cn_dataP)) {
   if (str->ordout.terror != RECOVERR)
     str->ordout.terror = IRRECERR;
   return (-1);
 }

 str->ordout.terror = NOERR;
 str->ordout.c_id = ordP->c_id;
 strncpy (str->ordout.c_last, ordP->c_last, 17);
 strncpy (str->ordout.c_first, ordP->c_first, 17);
 strncpy (str->ordout.c_middle, ordP->c_middle, 3);
 str->ordout.c_balance = ordP->c_balance/100;
 str->ordout.o_id = ordP->o_id;
 strncpy (str->ordout.o_entry_d, (char*)ordP->o_entry_d, 20);
 if ( ordP->o_carrier_id == 11 )
    str->ordout.o_carrier_id = 0;
 else
    str->ordout.o_carrier_id = ordP->o_carrier_id;
 str->ordout.o_ol_cnt = ordP->o_ol_cnt;
 for (i = 0; i < ordP->o_ol_cnt; i++) {
   ordP->ol_delivery_d[i][10] = '\0';
   if ( !strcmp((char*)ordP->ol_delivery_d[i],"01-01-1811"))
      strncpy((char*)ordP->ol_delivery_d[i],"NOTDELIVR",10);
   str->ordout.ol_supply_w_id[i] = ordP->ol_supply_w_id[i];
   str->ordout.ol_i_id[i] = ordP->ol_i_id[i];
   str->ordout.ol_quantity[i] = ordP->ol_quantity[i];
   str->ordout.ol_amount[i] = (float)(ordP->ol_amount[i])/100;
   strncpy (str->ordout.ol_delivery_d[i], (char*)ordP->ol_delivery_d[i], 11);
 }
 str->ordout.retry = ordP->retries;

#if defined(TOP) || defined(TUX)
 return(1);
#else
 return (0);
#endif

}

TPCdel (cnP, str)
void *cnP;
struct delstruct *str;

{
 ora_cn_data_t *cn_dataP = (ora_cn_data_t *)cnP;
 global_delivery_t*delP = cn_dataP->delP;
 OCIError *errhp = cn_dataP->errhp;
 double tr_end, tr_begin;
 int i, skipped;
 struct timeval cur_time;
 static int tran_cntr=0;
 int pos, len;
 int queue_time, start_time, end_time;
 char stdout_buf[1024];

 /*  Open the delivery log file if needed */
 if (lfp == NULL) {
   DVRY_LOCK;
   if (lfp == NULL) {
     err_printf("TPCdel: delivery_file_nameis %s\n", delivery_file_name);
     if ((lfp = fopen (delivery_file_name,"w")) == NULL) {
       fprintf (stderr, "Error in TPC-C server: Failed to open %s\n",
             delivery_file_name);
       DVRY_UNLOCK;
       return(-1);
     }
     err_printf("Opened delivery file%s\n", delivery_file_name);
   }
   DVRY_UNLOCK;
 }

#ifndef USE_ORACLE_DVRY_FORMAT
 gettimeofday(&cur_time,NULL);
 tr_begin = (double)cur_time.tv_sec + 1.0e-6 * (double)cur_time.tv_usec;
 start_time = cur_time.tv_sec;
#endif

 delP->w_id = str->delin.w_id;
 delP->o_carrier_id = str->delin.o_carrier_id;
 delP->retries = 0;
 delP->plsqlflag= 1;
/*
 vgetdate(cr_date);  */
 OCIERROR(errhp,OCIDateSysDate(errhp,&delP->cr_date));

  if (str->delout.terror = tkvcd (cn_dataP)) {
 /* if (str->delout.terror = tkvcd (delP->plsqlflag)) {   */
  if(str->delout.terror == DEL_ERROR)
    return DEL_ERROR;
  if (str->delout.terror != RECOVERR)
     str->delout.terror = IRRECERR;
  return (-1);
 }

#ifdef USE_ORACLE_DVRY_FORMAT
 tr_end = gettime ();
 DVRY_LOCK;
 fprintf (lfp, "%d %d %f %f %d %d", str->delin.in_timing_int,
          (tr_end - str->delin.qtime) <= DELRT ? 1 : 0,
          str->delin.qtime, tr_end, delP->w_id, delP->o_carrier_id);
 for (i = 0; i < 10; i++) {
   fprintf (lfp, " %d %d", i + 1, delP->del_o_id[i]);
   if (delP->del_o_id[i] <= 0) {
#ifdef TUX
      userlog ("DELIVERY: no new order for w_id: %d, d_id %d\n",
            delP->w_id, i + 1);
#else
      fprintf (stderr, "DELIVERY: no new order for w_id: %d, d_id %d\n",
            delP->w_id, i + 1);
#endif
   }
 }
 fprintf (lfp, " %d\n", delP->retries);

#else  /* not USE_ORACLE_DVRY_FORMAT */
 gettimeofday(&cur_time, NULL);
 tr_end = (double)cur_time.tv_sec + 1.0e-6 * (double)cur_time.tv_usec;
 end_time = cur_time.tv_sec;

 queue_time = str->delin.qtime;
 pos = 0;

 DVRY_LOCK;
 ++tran_cntr;
 pos += sprintf(&stdout_buf[pos], "--Tran %d Queue %.3f Start %.3f\n",
            tran_cntr, str->delin.qtime, tr_begin);
 pos += sprintf(&stdout_buf[pos], "W_ID: %d, CARRIER_ID: %d",
            str->delin.w_id, str->delin.o_carrier_id);

 if (str->delout.terror == DEL_ERROR) {
   pos += sprintf(&stdout_buf[pos],
            "\nDelivery transaction failed (DEL_ERROR)\n");
 } else if (str->delout.terror != 0) {
   pos += sprintf(&stdout_buf[pos], "Delivery transaction failed (%d)",
            str->delout.terror);
 } else {
   int skipped[10];
   int num_skipped = 0;
   for (i = 0; i < 10; i++) {
      if (delP->del_o_id[i] <= 0) {
         skipped[i] = 1;
         num_skipped++;
      } else {
         skipped[i] = 0;
      }
      pos += sprintf(&stdout_buf[pos], " %d", delP->del_o_id[i]);
   }
   pos += sprintf(&stdout_buf[pos], "\n");
   if (num_skipped > 0) {
      for (i=0; i<10; i++) {
         if (skipped[i] == 1) {
            pos += sprintf(&stdout_buf[pos],
                  "D_ID %d has no new orders.\n", i+1);
         }
      }
   }
 }

 fprintf(lfp, "%send-time: %.3f\n", stdout_buf, tr_end);
 fflush(lfp);
#endif /* USE_ORACLE_DVRY_FORMAT */

 DVRY_UNLOCK;
 str->delout.terror = NOERR;
 str->delout.retry = delP->retries;

#if defined(TOP) || defined(TUX)   /* changed mjb 17 feb */
 return(1);
#else
 return (0);
#endif

}

TPCsto (cnP, str)
void *cnP;
struct stostruct *str;

{
 ora_cn_data_t *cn_dataP = (ora_cn_data_t *)cnP;
 global_stock_t *stoP = cn_dataP->stoP;

 stoP->w_id = str->stoin.w_id;
 stoP->d_id = str->stoin.d_id;
 stoP->threshold = str->stoin.threshold;
 stoP->retries = 0;

 if (str->stoout.terror = tkvcs (cn_dataP)) {
   if (str->stoout.terror != RECOVERR)
     str->stoout.terror = IRRECERR;
   return (-1);
 }

 str->stoout.terror = NOERR;
 str->stoout.low_stock = stoP->low_stock;
 str->stoout.retry = stoP->retries;
```

```
#if defined(TOP) || defined(TUX)   /* changed mjb 17 feb */
  return(1);
#else
  return (0);
#endif

}

#ifndef AVOID_DEADLOCK

void q_sort(global_newOrder_t *newP,struct newstruct *str,int left, int right, int *indx, int in)
{
 int i, last;

 if(left >= right)
   return;
 swap(newP, str,left,(left+right)/2, indx, in);
 last = left;
 for(i=left+1;i<=right;i++)
   if(newP->nol_i_id[i]< newP->nol_i_id[left])
     swap(newP, str,last,i, indx, in);
 swap(newP, str,left,last, indx, in);
 q_sort(newP,str,left,last-1, indx, in);
 q_sort(newP,str,last+1,right, indx, in);
}

void swap(global_newOrder_t *newP, struct newstruct *str, int i, int j, int *indx, int in)
{
 int temp;
 char tmpstr[25];
 char tmpch;

 temp = indx[i];
 indx[i] = indx[j];
 indx[j] = temp;

 if (in) {

                temp = newP->nol_i_id[i];
                newP->nol_i_id[i] = newP->nol_i_id[j];
                newP->nol_i_id[j] = temp;
                temp = newP->nol_supply_w_id[i];
                newP->nol_supply_w_id[i]= newP->nol_supply_w_id[j];
                newP->nol_supply_w_id[j]= temp;

                temp = newP->nol_quantity[i];
                newP->nol_quantity[i] = newP->nol_quantity[j];
                newP->nol_quantity[j] = temp;
  } else {

                strcpy(tmpstr,str->newout.i_name[i]);
                strcpy(str->newout.i_name[i],str->newout.i_name[j]);
                strcpy(str->newout.i_name[j],tmpstr);

                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;

                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
```

views.sql

```
create or replace view wh_dist
(w_id, d_id, d_tax, d_next_o_id, w_tax )
as select w.w_id, d.d_id, d.d_tax, d.d_next_o_id, w.w_tax
  from dist d, ware w
  where w.w_id = d.d_w_id
/

create  or replace view stock_item
(i_id, s_w_id, i_price, i_name, i_data, s_data, s_quantity,
s_order_cnt, s_ytd, s_remote_cnt,
s_dist_01, s_dist_02, s_dist_03, s_dist_04, s_dist_05,
s_dist_06, s_dist_07, s_dist_08, s_dist_09, s_dist_10)
as
select 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
  /
```

APPENDIX B: Tunable Parameters

B.1 Database Parameters

```
compatible = 10.0.0.0.0
db_name = tpcc
control_files = /dev/rtpc_lvcntrl1
_column_compression_factor = 0
recovery_parallelism = 512
dml_locks = 500
log_buffer = 33554432
parallel_max_servers = 512

db_files = 2400
fast_start_io_target = 0

db_cache_size = 48256M
db_2k_cache_size = 512M
db_8k_cache_size = 912M
db_16k_cache_size = 52768M
db_keep_cache_size = 345488M
db_recycle_cache_size= 19600M
enqueue_resources = 60000
processes = 1600
sessions = 800
transactions = 790
shared_pool_size = 3768M
cursor_space_for_time = TRUE
db_block_size = 4096
undo_management = auto

UNDO_TABLESPACE = undo_ts

_ksmg_granule_size = 268435456
_db_cache_pre_warm=FALSE

trace_enabled = FALSE
db_block_checksum = FALSE
trace_enabled = FALSE
statistics_level = basic
timed_statistics = FALSE

plsql_optimize_level = 2
pga_aggregate_target = 0

undo_retention = 0
_imu_pools = 900
_optimizer_cache_stats = false
_optimizer_cost_model = io
fast_start_mttr_target = 0

db_writer_processes = 8
log_checkpoint_interval = 216500000
log_checkpoints_to_alert = TRUE
log_checkpoint_timeout = 1700
java_pool_size = 0
remote_login_passwordfile = shared
disk_asynch_io = TRUE
db_block_checking = FALSE
cursor_space_for_time = TRUE
lock_sga = TRUE
hash_join_enabled = FALSE
replication_dependency_tracking = FALSE
db_file_multiblock_read_count = 1
_cursor_cache_frame_bind_memory= true
max_dump_file_size=5M
_db_writer_coalesce_area_size = 0
aq_tm_processes = 0
_kghdsidx_count = 1
_second_spare_parameter = 1
_third_spare_parameter = 784
_two_pass=false
```

B.2 Transaction Monitor Parameters

Tpccrc

```
CellLogVolume ecmlog
CellDataVolume ecmdata
NodeLogVolume enmlog
TpccApplicationDirectory /home/encina
TpccDbServer oratpcc.world
StatsFrequency 10
Version 1.0
Servers:delivery PAS 1 Threads 3 Name del IFS ---D- Dvry 2
Servers:1online PAS 1 Threads 1 Name on1 IFS NPO-S Dvry 0
Servers:2online PAS 1 Threads 1 Name on2 IFS NPO-S Dvry 0
Servers:3online PAS 1 Threads 1 Name on3 IFS NPO-S Dvry 0
Servers:4online PAS 1 Threads 1 Name on4 IFS NPO-S Dvry 0
Servers:5online PAS 1 Threads 1 Name on5 IFS NPO-S Dvry 0
Servers:6online PAS 1 Threads 1 Name on6 IFS NPO-S Dvry 0
Servers:7online PAS 1 Threads 1 Name on7 IFS NPO-S Dvry 0
Servers:8online PAS 1 Threads 1 Name on8 IFS NPO-S Dvry 0
Servers:9online PAS 1 Threads 1 Name on9 IFS NPO-S Dvry 0
Servers:10online PAS 1 Threads 1 Name on10 IFS NPO-S Dvry 0
Servers:11online PAS 1 Threads 1 Name on11 IFS NPO-S Dvry 0
Servers:12online PAS 1 Threads 1 Name on12 IFS NPO-S Dvry 0
Servers:13online PAS 1 Threads 1 Name on13 IFS NPO-S Dvry 0
Servers:14online PAS 1 Threads 1 Name on14 IFS NPO-S Dvry 0
Servers:15online PAS 1 Threads 1 Name on15 IFS NPO-S Dvry 0
Servers:16online PAS 1 Threads 1 Name on16 IFS NPO-S Dvry 0
Servers:17online PAS 1 Threads 1 Name on17 IFS NPO-S Dvry 0
Servers:18online PAS 1 Threads 1 Name on18 IFS NPO-S Dvry 0
Servers:19online PAS 1 Threads 1 Name on19 IFS NPO-S Dvry 0
Servers:20online PAS 1 Threads 1 Name on20 IFS NPO-S Dvry 0
Servers:21online PAS 1 Threads 1 Name on21 IFS NPO-S Dvry 0
Servers:22online PAS 1 Threads 1 Name on22 IFS NPO-S Dvry 0
Servers:23online PAS 1 Threads 1 Name on23 IFS NPO-S Dvry 0
Servers:24online PAS 1 Threads 1 Name on24 IFS NPO-S Dvry 0
```

B.3 AIX Parameters

IBM eServer pSeries 690

```
                OS PARAMETERS
keylock     normal      State of system keylock at boot time          False
maxbuf      20          Maximum number of pages in block I/O BUFFER CACHE True
maxmbuf     0           Maximum Kbytes of real memory allowed for MBUFS   True
maxuproc    40000       Maximum number of PROCESSES allowed per user      True
autorestart true        Automatically REBOOT system after a crash         True
iostat      false       Continuously maintain DISK I/O history            True
realmem     536870912   Amount of usable physical memory in Kbytes        False
conslogin   enable      System Console Login                              False
fwversion   IBM,RG030510 Firmware version and revision levels             False
maxpout     0           HIGH water mark for pending write I/Os per file   True
minpout     0           LOW water mark for pending write I/Os per file    True
fullcore    false       Enable full CORE dump                             True
pre430core  false       Use pre-430 style CORE dump                       True
ncargs      20          ARG/ENV list size in 4K byte blocks               True
pre520tune  disable     Pre-520 tuning compatibility mode                 True
rtasversion 1           Open Firmware RTAS version                        False
modelname   IBM,7040-681 Machine name                                     False
systemid    IBM,010YHRH0G Hardware system identifier                      False
boottype    disk        N/A                                               False
SW_dist_intr false      Enable SW distribution of interrupts              True
cpuguard    enable      CPU Guard                                         True
frequency   0           System Bus Frequency                              False


vmo -o v_pinshm=1
vmo -r -o lgpg_size=16777216 -o lgpg_regions=31010
vmo -o htabscale=-4
chuser capabilities=CAP_BYPASS_RAC_VMM,CAP_PROPAGATE  oracle
MEMORY_AFFINITY=MCM
REPLICATE_TEXT=/home/oracle/bin/oracle
DATA_SEG_SPECIAL=Y
```

Appendix C: Database Setup Code

C.1 Database Creation Scripts

createuser.sh

```
#!/usr/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
```

createts.sh

```
$tpcc_createts /dev/rtpc_lvtemp temp 140 1       6399M 50K  unix 1       0 32 auto t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for temp failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lvi2cu icust2 29 1       3519M 175M  unix 0       1863 32 auto t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for icust2 failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lvcust cust 583 1       3455M 1727M  unix 0       1 32 auto t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for cust failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lvware ware 1 1       159M 1M  unix 0       0 32 2K t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for ware failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lvdist dist 1 1       1343M 1M  unix 0       584 32 2K t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for dist failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lvhist hist 24 1       9663M 483M  unix 0       585 32 8K t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for hist failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lvstk stok 1015 1       2399M 1199M  unix 0       609 32 auto t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for stok failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lvitem item 1 1       63M 2M  unix 0       1624 32 2K t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for item failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lvordl ordr 216 1       12831M 641M  unix 0       1625 32 16K t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for ordr failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lvnord nord 12 1       1439M 143M  unix 0       1841 32 auto t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for nord failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lviware iware 1 1       95M 1M  unix 0       1853 32 2K t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for iware failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lvi1cu icust1 9 1       6111M 305M  unix 0       1854 32 auto t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for icust1 failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lvidist idist 1 1       319M 1M  unix 0       1890 32 2K t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for idist failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lvistk istok 18 1       10399M 519M  unix 0       1891 32 auto t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for istok failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lviitem iitem 1 1       2560K 20K  unix 0       1909  32 2K t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for iitem failed.  Exiting.
    exit 0
  fi


$tpcc_createts /dev/rtpc_lvi2or iordr2 126 1       1119M 55M  unix 0       1910 32 auto t
  if expr $? != 0 > /dev/null; then
    echo Creating tablespace for iordr2 failed.  Exiting.
    exit 0
  fi
```

addfile.sh

```
#!/bin/sh
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

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
   autospace=auto
 else
```

```
  autospace=manual
 fi
  createsql="create tablespace $1 datafile '$2' size $3 reuse extent management local uniform size $4
segment space management $autospace $bssql nologging ;"
fi

$tpcc_sqlplus $tpcc_user_pass <<!
 spool createts_$1.log
 set echo on
 set timing on
 drop tablespace $1 including contents;
$createsql
 set echo off
 spool off
 exit ;
!
```

assigntemp.sh

```
#!/bin/sh

echo Assigning temporary tablespace to user tpcc...
$tpcc_sqlplus $tpcc_dba_user_pass @$tpcc_sql_dir/assigntemp> junk 2>&1
if test $? -ne 0
then
 exit 1;
else
 exit 0;
fi
```

stepenv.sh

```
# forces any env variables we set to be exported
set -a
tpcc_kit=t
tpcc_bench=$PWD
tpcc_scripts=$tpcc_bench/scripts
tpcc_require=$tpcc_scripts/require_vars.sh
tpcc_lcm=$tpcc_scripts/lcm.sh
tpcc_tokilobytes=$tpcc_scripts/tokilobytes.sh
tpcc_fromkilobytes=$tpcc_scripts/fromkilobytes.sh
tpcc_estsize=$tpcc_scripts/estsize.sh
tpcc_notneg=$tpcc_scripts/notneg.sh
tpcc_isneg=$tpcc_scripts/isneg.sh
tpcc_bcexpr=$tpcc_scripts/bcexpr.sh

if test -x /bin/ksh; then
tpcc_createts=$tpcc_scripts/createts.ksh
else
tpcc_createts=$tpcc_scripts/createts.sh
fi

tpcc_tabledata=$tpcc_scripts/tabledata.sh
tpcc_load=$tpcc_bench/benchrun/bin/tpccload.exe
tpcc_createtablespaces=$tpcc_scripts/createtablespaces.sh

tpcc_sqlplus=cat
tpcc_sqlplus_args='/nolog'
tpcc_internal_connect='connect / as sysdba'
tpcc_user_pass='tpcc/tpcc'
tpcc_dba_user_pass='system/manager'
oracle_dba=system
oracle_dba_password=manager
tpcc_sqlplus_args=
tpcc_user_pass=
tpcc_sqlplus=sqlplus
tpcc_user_pass='tpcc/tpcc'

. ${tpcc_bench}/options.sh

tpcc_fsize_limit_k=8388608
tpcc_extent_limit_k=2097151
tpcc_runlen=`$tpcc_bcexpr 8 \* 60 \* $tpcc_runlen`

tpcc_system_size=200M
tpcc_logfile_size=`$tpcc_bcexpr 20 + \( $tpcc_scale \)`M

tpcc_undo_size=`$tpcc_bcexpr 2 \* $tpcc_scale`M
tpcc_undo_bs=8K

tpcc_statspack_size=`$tpcc_bcexpr 1 \* $tpcc_scale`M
tpcc_sysaux_size=120


tpcc_table_list='ware cust dist hist stok item ordr ordl nord'
tpcc_index_list='iware icust1 icust2 idist istok iitem iordr1 iordr2 iordl inord'
tpcc_hist_growth=51
tpcc_ordr_growth=35
tpcc_nord_growth=13
#tpcc_ordl_growth=660
tpcc_ordl_growth=900

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

tpcc_tempts_min=10240

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
for table in $tpcc_table_list $tpcc_index_list temp; do
  eval "tpcc_${table}_tsfileinc="
 done
 tpcc_stok_tsfileinc=
 tpcc_cust_tsfileinc=
 tpcc_iordl2_tsfileinc=
 tpcc_icust2_tsfileinc=
 tpcc_iordl_tsfileinc=
fi


. ${tpcc_bench}/localoptions.sh

if expr `echo x$tpcc_no_options` = xt > /dev/null; then
 echo Please modify ${tpcc_bench}/localoptions.sh to configure the generator.
 exit 1
fi

tpcc_fixordrordl=${tpcc_genscripts_dir}/loadfixordrordl.sh
tpcc_updateordrordl=${tpcc_scripts}/updateordrordl.sh

tp(){
 eval echo \"'"\$tpcc_$1_$2"'\"
}

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

if test x$tpcc_os = xnt; then
 tpcc_hardcode=t
else
 tpcc_hardcode=f
fi

if test x$tpcc_defbs = x; then
 tpcc_defbs=2
fi

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="buildcreatets buildcreatedb \
buildcreatetable-ware buildcreatetable-cust buildcreatetable-dist buildcreatetable-hist buildcreatetable-stok
buildcreatetable-item buildcreatetable-ordr buildcreatetable-ordl buil
```

```
dcreatetable-nord \
buildloadware buildloaddist 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\
listfiles
"

tpcc_steps="runsqllocal-createdb shutdowndb 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-cre
atetable_ordl runsql-createtable_nord \
runscript-loadware runscript-loaddist 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 r
unsql-createindex_iordr2 runsql-createindex_iordl runsql-createindex_inord\
analyze runscript-loadfixordrordl createstats createstoredprocs createspacestats createmisc"

set +a
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

if $tpcc_require ORACLE_SID \
 tpcc_tokilobytes tpcc_createts tpcc_lcm\
 tpcc_sqlplus tpcc_internal_connect\
 tpcc_np tpcc_cpu tpcc_os tpcc_runlen tpcc_ldrive tpcc_scale tpcc_disks_location tpcc_auto_undo
tpcc_tempts_min\
 tpcc_system_size tpcc_logfile_size\
 tpcc_undo_size tpcc_undo_bs\
 oracle_dba oracle_dba_password tpcc_dba_user_pass
then exit 1; fi
```

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 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 listfilesonly)"
  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

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$controlcmd" = x; then
  endstep=
elif test "x$controlcmd" = xonly; then
  controlcmd=only
elif test "x$controlcmd" = xthrough; then
  actualstep=f
  for step in $tpcc_create_steps $tpcc_steps ; do
    if test "x$step" = "x$endstep"; then
      actualstep=t
    fi
  done
  if test $actualstep = f; then
    echo "Invalid step $endstep.  Use $0 steps  to show steps."
    exit 1
  fi
else
  echo "Invalid syntax.  Use $0 by itself for help."
  exit 1
fi

echo Starting from step: $startstep

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

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

loadware.sh

```
cd $tpcc_bench
$tpcc_load -M $tpcc_scale -w > loadware.log 2>&1
```

loaddist.sh

```
cd $tpcc_bench
$tpcc_load -M $tpcc_scale -d > loaddist.log 2>&1
```

loaditem.sh

```
cd $tpcc_bench
$tpcc_load -M $tpcc_scale -i > loaditem.log 2>&1
```

loadhist.sh

```
rm loadhist*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 65000 -h  -b 1 -e 1015 >> loadhist0.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 1016 -e 2030 >> loadhist1.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 2031 -e 3045 >> loadhist2.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 3046 -e 4060 >> loadhist3.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 4061 -e 5075 >> loadhist4.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 5076 -e 6090 >> loadhist5.log 2>&1 &
```

```
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 6091 -e 7105 >> loadhist6.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 7106 -e 8120 >> loadhist7.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 8121 -e 9135 >> loadhist8.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 9136 -e 10150 >> loadhist9.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 10151 -e 11165 >> loadhist10.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 11166 -e 12180 >> loadhist11.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 12181 -e 13195 >> loadhist12.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 13196 -e 14210 >> loadhist13.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 14211 -e 15225 >> loadhist14.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 15226 -e 16240 >> loadhist15.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 16241 -e 17255 >> loadhist16.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 17256 -e 18270 >> loadhist17.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 18271 -e 19285 >> loadhist18.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 19286 -e 20300 >> loadhist19.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 20301 -e 21315 >> loadhist20.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 21316 -e 22330 >> loadhist21.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 22331 -e 23345 >> loadhist22.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 23346 -e 24360 >> loadhist23.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 24361 -e 25376 >> loadhist24.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 25377 -e 26392 >> loadhist25.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 26393 -e 27408 >> loadhist26.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 27409 -e 28424 >> loadhist27.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 28425 -e 29440 >> loadhist28.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 29441 -e 30456 >> loadhist29.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 30457 -e 31472 >> loadhist30.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 31473 -e 32488 >> loadhist31.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 32489 -e 33504 >> loadhist32.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 33505 -e 34520 >> loadhist33.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 34521 -e 35536 >> loadhist34.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 35537 -e 36552 >> loadhist35.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 36553 -e 37568 >> loadhist36.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 37569 -e 38584 >> loadhist37.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 38585 -e 39600 >> loadhist38.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 39601 -e 40616 >> loadhist39.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 40617 -e 41632 >> loadhist40.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 41633 -e 42648 >> loadhist41.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 42649 -e 43664 >> loadhist42.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 43665 -e 44680 >> loadhist43.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 44681 -e 45696 >> loadhist44.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 45697 -e 46712 >> loadhist45.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 46713 -e 47728 >> loadhist46.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 47729 -e 48744 >> loadhist47.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 48745 -e 49760 >> loadhist48.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 49761 -e 50776 >> loadhist49.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 50777 -e 51792 >> loadhist50.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 51793 -e 52808 >> loadhist51.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 52809 -e 53824 >> loadhist52.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 53825 -e 54840 >> loadhist53.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 54841 -e 55856 >> loadhist54.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 55857 -e 56872 >> loadhist55.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 56873 -e 57888 >> loadhist56.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 57889 -e 58904 >> loadhist57.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 58905 -e 59920 >> loadhist58.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 59921 -e 60936 >> loadhist59.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 60937 -e 61952 >> loadhist60.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 61953 -e 62968 >> loadhist61.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 62969 -e 63984 >> loadhist62.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -h  -b 63985 -e 65000 >> loadhist63.log 2>&1 &
allprocs="$allprocs${!}"
error=0
for curproc in $allprocs; do
 wait $curproc
 error=`expr $? + $error`
done
exit `expr $error != 0`
```

loadnord.sh

```
rm loadnord*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 65000 -n  -b 1 -e 1015 >> loadnord0.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 1016 -e 2030 >> loadnord1.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 2031 -e 3045 >> loadnord2.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 3046 -e 4060 >> loadnord3.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 4061 -e 5075 >> loadnord4.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 5076 -e 6090 >> loadnord5.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 6091 -e 7105 >> loadnord6.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 7106 -e 8120 >> loadnord7.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 8121 -e 9135 >> loadnord8.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 9136 -e 10150 >> loadnord9.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 10151 -e 11165 >> loadnord10.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 11166 -e 12180 >> loadnord11.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 12181 -e 13195 >> loadnord12.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 13196 -e 14210 >> loadnord13.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 14211 -e 15225 >> loadnord14.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 15226 -e 16240 >> loadnord15.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 16241 -e 17255 >> loadnord16.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 17256 -e 18270 >> loadnord17.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 18271 -e 19285 >> loadnord18.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 19286 -e 20300 >> loadnord19.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 20301 -e 21315 >> loadnord20.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 21316 -e 22330 >> loadnord21.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 22331 -e 23345 >> loadnord22.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 23346 -e 24360 >> loadnord23.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 24361 -e 25376 >> loadnord24.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 25377 -e 26392 >> loadnord25.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 26393 -e 27408 >> loadnord26.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 27409 -e 28424 >> loadnord27.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 28425 -e 29440 >> loadnord28.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 29441 -e 30456 >> loadnord29.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 30457 -e 31472 >> loadnord30.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 31473 -e 32488 >> loadnord31.log 2>&1 &
allprocs="$allprocs${!}"
```

```
$tpcc_load -M 65000 -n  -b 32489 -e 33504 >> loadnord32.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 33505 -e 34520 >> loadnord33.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 34521 -e 35536 >> loadnord34.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 35537 -e 36552 >> loadnord35.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 36553 -e 37568 >> loadnord36.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 37569 -e 38584 >> loadnord37.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 38585 -e 39600 >> loadnord38.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 39601 -e 40616 >> loadnord39.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 40617 -e 41632 >> loadnord40.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 41633 -e 42648 >> loadnord41.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 42649 -e 43664 >> loadnord42.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 43665 -e 44680 >> loadnord43.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 44681 -e 45696 >> loadnord44.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 45697 -e 46712 >> loadnord45.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 46713 -e 47728 >> loadnord46.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 47729 -e 48744 >> loadnord47.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 48745 -e 49760 >> loadnord48.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 49761 -e 50776 >> loadnord49.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 50777 -e 51792 >> loadnord50.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 51793 -e 52808 >> loadnord51.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 52809 -e 53824 >> loadnord52.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 53825 -e 54840 >> loadnord53.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 54841 -e 55856 >> loadnord54.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 55857 -e 56872 >> loadnord55.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 56873 -e 57888 >> loadnord56.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 57889 -e 58904 >> loadnord57.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 58905 -e 59920 >> loadnord58.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 59921 -e 60936 >> loadnord59.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 60937 -e 61952 >> loadnord60.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 61953 -e 62968 >> loadnord61.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 62969 -e 63984 >> loadnord62.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -n  -b 63985 -e 65000 >> loadnord63.log 2>&1 &
allprocs="$allprocs${!}"
error=0
for curproc in $allprocs; do
 wait $curproc
 error=`expr $? + $error`
done
exit `expr $error != 0`
```

loadordrordl.sh

```
rm loadordrordl*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy0.dat -b 1 -e 1015 >>
loadordrordl0.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy1.dat -b 1016 -e 2030
>> loadordrordl1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o  $tpcc_disks_location/dummy2.dat -b 2031 -e 3045
>> loadordrordl2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy3.dat -b 3046 -e 4060
>> loadordrordl3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy4.dat -b 4061 -e 5075
>> loadordrordl4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy5.dat -b 5076 -e  6090
>> loadordrordl5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy6.dat -b 6091 -e 7105
>> loadordrordl6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy7.dat -b 7106 -e 8120
>> loadordrordl7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy8.dat -b 8121 -e 9135
>> loadordrordl8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy9.dat -b 9136 -e 10150
>> loadordrordl9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy10.dat -b 10151 -e
11165 >> loadordrordl10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy11.dat -b 11166 -e
12180 >> loadordrordl11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy12.dat -b 12181 -e
13195 >> loadordrordl12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy13.dat -b 13196 -e
14210 >> loadordrordl13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy14.dat -b 14211 -e
15225 >> loadordrordl14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy15.dat -b 15226 -e
16240 >> loadordrordl15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy16.dat -b 16241 -e
17255 >> loadordrordl16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy17.dat -b 17256 -e
18270 >> loadordrordl17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy18.dat -b 18271 -e
19285 >> loadordrordl18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy19.dat -b 19286 -e
20300 >> loadordrordl19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy20.dat -b 20301 -e
21315 >> loadordrordl20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy21.dat -b 21316 -e
22330 >> loadordrordl21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy22.dat -b 22331 -e
23345 >> loadordrordl22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy23.dat -b 23346 -e
24360 >> loadordrordl23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy24.dat -b 24361 -e
25376 >> loadordrordl24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy25.dat -b 25377 -e
26392 >> loadordrordl25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy26.dat -b 26393 -e
27408 >> loadordrordl26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy27.dat -b 27409 -e
28424 >> loadordrordl27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy28.dat -b 28425 -e
29440 >> loadordrordl28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy29.dat -b 29441 -e
30456 >> loadordrordl29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy30.dat -b 30457 -e
31472 >> loadordrordl30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy31.dat -b 31473 -e
32488 >> loadordrordl31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy32.dat -b 32489 -e
33504 >> loadordrordl32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy33.dat -b 33505 -e
34520 >> loadordrordl33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy34.dat -b 34521 -e
35536 >> loadordrordl34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy35.dat -b 35537 -e
36552 >> loadordrordl35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy36.dat -b 36553 -e
37568 >> loadordrordl36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy37.dat -b 37569 -e
38584 >> loadordrordl37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy38.dat -b 38585 -e
39600 >> loadordrordl38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy39.dat -b 39601 -e
40616 >> loadordrordl39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy40.dat -b 40617 -e
41632 >> loadordrordl40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy41.dat -b 41633 -e
42648 >> loadordrordl41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy42.dat -b 42649 -e
43664 >> loadordrordl42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy43.dat -b 43665 -e
```

```
44680 >> loadordrordl43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy44.dat -b 44681 -e
45696 >> loadordrordl44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy45.dat -b 45697 -e
46712 >> loadordrordl45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy46.dat -b 46713 -e
47728 >> loadordrordl46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy47.dat -b 47729 -e
48744 >> loadordrordl47.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy48.dat -b 48745 -e
49760 >> loadordrordl48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy49.dat -b 49761 -e
50776 >> loadordrordl49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy50.dat -b 50777 -e
51792 >> loadordrordl50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy51.dat -b 51793 -e
52808 >> loadordrordl51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy52.dat -b 52809 -e
53824 >> loadordrordl52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy53.dat -b 53825 -e
54840 >> loadordrordl53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy54.dat -b 54841 -e
55856 >> loadordrordl54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy55.dat -b 55857 -e
56872 >> loadordrordl55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy56.dat -b 56873 -e
57888 >> loadordrordl56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy57.dat -b 57889 -e
58904 >> loadordrordl57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy58.dat -b 58905 -e
59920 >> loadordrordl58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy59.dat -b 59921 -e
60936 >> loadordrordl59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy60.dat -b 60937 -e
61952 >> loadordrordl60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o  $tpcc_disks_location/dummy61.dat -b 61953 -e
62968 >> loadordrordl61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy62.dat -b 62969 -e
63984 >> loadordrordl62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_load -M 65000 -o $tpcc_disks_location/dummy63.dat -b 63985 -e
65000 >> loadordrordl63.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
 wait $curproc
 error=`expr $? + $error`
done
exit `expr $error != 0`
```

loadfixordrordl.sh

```
rm loadfixordrordl*.log
cd $tpcc_bench
allprocs=
$tpcc_updateordrordl -M 65000 -o  -b 1 -e 1015 >> loadfixordrordl0.log
2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 1016 -e 2030 >>
loadfixordrordl1.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 2031 -e 3045 >>
loadfixordrordl2.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 3046 -e 4060 >>
loadfixordrordl3.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 4061 -e 5075 >>
loadfixordrordl4.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 5076 -e 6090 >>
loadfixordrordl5.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 6091 -e 7105 >>
loadfixordrordl6.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 7106 -e 8120 >>
loadfixordrordl7.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 8121 -e 9135 >>
loadfixordrordl8.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 9136 -e 10150 >>
loadfixordrordl9.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 10151 -e 11165 >>
loadfixordrordl10.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 11166 -e 12180 >>
loadfixordrordl11.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 12181 -e 13195 >>
loadfixordrordl12.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 13196 -e 14210 >>
loadfixordrordl13.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 14211 -e 15225 >>
loadfixordrordl14.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 15226 -e 16240 >>
loadfixordrordl15.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 16241 -e 17255 >>
loadfixordrordl16.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 17256 -e 18270 >>
loadfixordrordl17.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 18271 -e 19285 >>
loadfixordrordl18.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 19286 -e 20300 >>
loadfixordrordl19.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 20301 -e 21315 >>
loadfixordrordl20.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 21316 -e 22330 >>
loadfixordrordl21.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 22331 -e 23345 >>
loadfixordrordl22.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 23346 -e 24360 >>
loadfixordrordl23.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 24361 -e 25376 >>
loadfixordrordl24.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 25377 -e 26392 >>
loadfixordrordl25.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 26393 -e 27408 >>
loadfixordrordl26.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 27409 -e 28424 >>
loadfixordrordl27.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 28425 -e 29440 >>
loadfixordrordl28.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 29441 -e 30456 >>
loadfixordrordl29.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 30457 -e 31472 >>
loadfixordrordl30.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 31473 -e 32488 >>
loadfixordrordl31.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 32489 -e 33504 >>
loadfixordrordl32.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 33505 -e 34520 >>
loadfixordrordl33.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 34521 -e 35536 >>
loadfixordrordl34.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 35537 -e 36552 >>
loadfixordrordl35.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 36553 -e 37568 >>
loadfixordrordl36.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 37569 -e 38584 >>
loadfixordrordl37.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 38585 -e 39600 >>
loadfixordrordl38.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 39601 -e 40616 >>
loadfixordrordl39.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 40617 -e 41632 >>
loadfixordrordl40.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 41633 -e 42648 >>
loadfixordrordl41.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 42649 -e 43664 >>
loadfixordrordl42.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 43665 -e 44680 >>
loadfixordrordl43.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 44681 -e 45696 >>
loadfixordrordl44.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 45697 -e 46712 >>
loadfixordrordl45.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 46713 -e 47728 >>
loadfixordrordl46.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 47729 -e 48744 >>
loadfixordrordl47.log 2>&1 &
```

```
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 48745 -e 49760 >>
loadfixordrordl48.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 49761 -e 50776 >>
loadfixordrordl49.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 50777 -e 51792 >>
loadfixordrordl50.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 51793 -e 52808 >>
loadfixordrordl51.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 52809 -e 53824 >>
loadfixordrordl52.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 53825 -e 54840 >>
loadfixordrordl53.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 54841 -e 55856 >>
loadfixordrordl54.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 55857 -e 56872 >>
loadfixordrordl55.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 56873 -e 57888 >>
loadfixordrordl56.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 57889 -e 58904 >>
loadfixordrordl57.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 58905 -e 59920 >>
loadfixordrordl58.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 59921 -e 60936 >>
loadfixordrordl59.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 60937 -e 61952 >>
loadfixordrordl60.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 61953 -e 62968 >>
loadfixordrordl61.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 62969 -e 63984 >>
loadfixordrordl62.log 2>&1 &
allprocs="$allprocs ${!}"
$tpcc_updateordrordl -M 65000 -o  -b 63985 -e 65000 >>
loadfixordrordl63.log 2>&1 &
allprocs="$allprocs ${!}"
error=0
for curproc in $allprocs; do
 wait $curproc
 error=`expr $? + $error`
done
exit `expr $error != 0`
```

loadcust.sh

```
rm loadcust*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 65000 -c  -b 1 -e 1015 >> loadcust0.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 1016 -e 2030 >> loadcust1.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 2031 -e 3045 >> loadcust2.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 3046 -e 4060 >> loadcust3.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 4061 -e 5075 >> loadcust4.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 5076 -e 6090 >> loadcust5.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 6091 -e 7105 >> loadcust6.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 7106 -e 8120 >> loadcust7.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 8121 -e 9135 >> loadcust8.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 9136 -e 10150 >> loadcust9.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 10151 -e 11165 >> loadcust10.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 11166 -e 12180 >> loadcust11.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 12181 -e 13195 >> loadcust12.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 13196 -e 14210 >> loadcust13.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 14211 -e 15225 >> loadcust14.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 15226 -e 16240 >> loadcust15.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 16241 -e 17255 >> loadcust16.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 17256 -e 18270 >> loadcust17.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 18271 -e 19285 >> loadcust18.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 19286 -e 20300 >> loadcust19.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 20301 -e 21315 >> loadcust20.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 21316 -e 22330 >> loadcust21.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 22331 -e 23345 >> loadcust22.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 23346 -e 24360 >> loadcust23.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 24361 -e 25376 >> loadcust24.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 25377 -e 26392 >> loadcust25.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 26393 -e 27408 >> loadcust26.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 27409 -e 28424 >> loadcust27.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 28425 -e 29440 >> loadcust28.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 29441 -e 30456 >> loadcust29.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 30457 -e 31472 >> loadcust30.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 31473 -e 32488 >> loadcust31.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 32489 -e 33504 >> loadcust32.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 33505 -e 34520 >> loadcust33.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 34521 -e 35536 >> loadcust34.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 35537 -e 36552 >> loadcust35.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 36553 -e 37568 >> loadcust36.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 37569 -e 38584 >> loadcust37.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 38585 -e 39600 >> loadcust38.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 39601 -e 40616 >> loadcust39.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 40617 -e 41632 >> loadcust40.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 41633 -e 42648 >> loadcust41.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 42649 -e 43664 >> loadcust42.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 43665 -e 44680 >> loadcust43.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 44681 -e 45696 >> loadcust44.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 45697 -e 46712 >> loadcust45.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 46713 -e 47728 >> loadcust46.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 47729 -e 48744 >> loadcust47.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 48745 -e 49760 >> loadcust48.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 49761 -e 50776 >> loadcust49.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 50777 -e 51792 >> loadcust50.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 51793 -e 52808 >> loadcust51.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 52809 -e 53824 >> loadcust52.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 53825 -e 54840 >> loadcust53.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 54841 -e 55856 >> loadcust54.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 55857 -e 56872 >> loadcust55.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 56873 -e 57888 >> loadcust56.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 57889 -e 58904 >> loadcust57.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 58905 -e 59920 >> loadcust58.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 59921 -e 60936 >> loadcust59.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 60937 -e 61952 >> loadcust60.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 61953 -e 62968 >> loadcust61.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 62969 -e 63984 >> loadcust62.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -c  -b 63985 -e 65000 >> loadcust63.log 2>&1 &
allprocs="$allprocs${!}"
error=0
for curproc in $allprocs; do
 wait $curproc
 error=`expr $? + $error`
done
exit `expr $error != 0`
```

loadstok.sh

```
rm loadstok*.log
cd $tpcc_bench
allprocs=
$tpcc_load -M 65000 -S  -j 1 -k 1562 >> loadstok0.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 1563 -k 3124 >> loadstok1.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 3125 -k 4686 >> loadstok2.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 4687 -k 6248 >> loadstok3.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 6249 -k 7810 >> loadstok4.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 7811 -k 9372 >> loadstok5.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 9373 -k 10934 >> loadstok6.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 10935 -k 12496 >> loadstok7.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 12497 -k 14058 >> loadstok8.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 14059 -k 15620 >> loadstok9.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 15621 -k 17182 >> loadstok10.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 17183 -k 18744 >> loadstok11.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 18745 -k 20306 >> loadstok12.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 20307 -k 21868 >> loadstok13.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 21869 -k 23430 >> loadstok14.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 23431 -k 24992 >> loadstok15.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 24993 -k 26554 >> loadstok16.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 26555 -k 28116 >> loadstok17.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 28117 -k 29678 >> loadstok18.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 29679 -k 31240 >> loadstok19.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 31241 -k 32802 >> loadstok20.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 32803 -k 34364 >> loadstok21.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 34365 -k 35926 >> loadstok22.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 35927 -k 37488 >> loadstok23.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 37489 -k 39050 >> loadstok24.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 39051 -k 40612 >> loadstok25.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 40613 -k 42174 >> loadstok26.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 42175 -k 43736 >> loadstok27.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 43737 -k 45298 >> loadstok28.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 45299 -k 46860 >> loadstok29.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 46861 -k 48422 >> loadstok30.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 48423 -k 49984 >> loadstok31.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 49985 -k 51547 >> loadstok32.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 51548 -k 53110 >> loadstok33.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 53111 -k 54673 >> loadstok34.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 54674 -k 56236 >> loadstok35.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 56237 -k 57799 >> loadstok36.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 57800 -k 59362 >> loadstok37.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 59363 -k 60925 >> loadstok38.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 60926 -k 62488 >> loadstok39.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 62489 -k 64051 >> loadstok40.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 64052 -k 65614 >> loadstok41.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 65615 -k 67177 >> loadstok42.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 67178 -k 68740 >> loadstok43.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 68741 -k 70303 >> loadstok44.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 70304 -k 71866 >> loadstok45.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 71867 -k 73429 >> loadstok46.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 73430 -k 74992 >> loadstok47.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 74993 -k 76555 >> loadstok48.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 76556 -k 78118 >> loadstok49.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 78119 -k 79681 >> loadstok50.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 79682 -k 81244 >> loadstok51.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 81245 -k 82807 >> loadstok52.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 82808 -k 84370 >> loadstok53.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 84371 -k 85933 >> loadstok54.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 85934 -k 87496 >> loadstok55.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 87497 -k 89059 >> loadstok56.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 89060 -k 90622 >> loadstok57.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 90623 -k 92185 >> loadstok58.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 92186 -k 93748 >> loadstok59.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 93749 -k 95311 >> loadstok60.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 95312 -k 96874 >> loadstok61.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 96875 -k 98437 >> loadstok62.log 2>&1 &
allprocs="$allprocs${!}"
$tpcc_load -M 65000 -S  -j 98438 -k 100000 >> loadstok63.log 2>&1 &
allprocs="$allprocs${!}"
error=0
for curproc in $allprocs; do
 wait $curproc
 error=`expr $? + $error`
done
exit `expr $error != 0`
```

C.2 SQL Scripts

createdb.sql

```
spool createdb.log
set echo on
shutdown abort

startup pfile=p_create.ora nomount
create database tpcc
 controlfile reuse
 maxinstances 1
 datafile '/dev/rtpc_lvsystem1' size 895M reuse
 logfile '/dev/rtpc_lvbblog1' size 605000M reuse,
         '/dev/rtpc_lvbblog2' size 605000M reuse
 sysaux datafile '/dev/rtpc_lvaux1' size 895M reuse ;

create undo tablespace undo_ts datafile
  '/dev/rtpc_lvroll1' size 6143M reuse blocksize 8K;

set echo off
exit sql.sqlcode
```

createuser.sql

```
spool createusertpcc.log;
set echo on;
create user tpcc identified by tpcc;
grant dba to tpcc;
set echo off;
spool off;
exit ;
```

assigntemp.sql

```
spool assigntemp.log;
set echo on;
alter user tpcc temporary tablespace temp_0;
set echo off;
spool off;
exit ;
```

createtable_ware.sql

```
set timing on
  set sqlblanklineson
  spool createtable_ware.log
  set echo on
    drop cluster warecluster including tables ;

create cluster warecluster (
 w_id number(5,0)
)
 single table
 hashkeys 65000
 hash is ( (w_id) )
 size 1536
   initrans 2
 storage ( buffer_pool default )
 tablespace ware_0;

create table ware (
 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)
)
cluster warecluster (
 w_id
);
   set echo off
   spool off
   exit sql.sqlcode;
```

createtable_cust.sql

```
set timing on
   set sqlblanklineson
   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 1950000000
 hash is ( (c_id * ( 65000 * 10 ) + c_w_id * 10 + c_d_id) )
 size 350
 pctfree 0  initrans 3
 storage ( buffer_pool keep )
 tablespace cust_0;

create table cust (
 c_id number
, c_d_id number
, c_w_id number
, c_discount number
, c_credit char(2)
, c_last varchar2(16)
, c_first varchar2(16)
, c_credit_lim number
, c_balance number
, c_ytd_payment number
, c_payment_cnt number
, c_delivery_cnt number
, c_street_1 varchar2(20)
, c_street_2 varchar2(20)
, c_city varchar2(20)
, c_state char(2)
, c_zip char(9)
, c_phone char(16)
, c_since date
, c_middle char(2)
, c_data varchar2(500)
)
cluster custcluster (
 c_id
, c_d_id
 , c_w_id
 );
   set echo off
   spool off
   exit sql.sqlcode;
```

createtable_dist.sql

```
set timing on
   set sqlblanklineson
   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 650000
 hash is ( ((d_w_id * 10) + d_id) )
 size 170
   initrans 4
 storage ( buffer_pool default )
 tablespace dist_0;

create table dist (
 d_id number
, d_w_id number
, d_ytd number
, d_next_o_id number
, d_tax number
, d_name varchar2(10)
, d_street_1 varchar2(20)
, d_street_2 varchar2(20)
, d_city varchar2(20)
, d_state char(2)
, d_zip char(9)
)
cluster distcluster (
 d_id
, d_w_id
);
   set echo off
   spool off
   exit sql.sqlcode;
```

createtable_histsql

```
set timing on
   set sqlblanklineson
   spool createtable_hist.log
   set echo on
     drop table hist ;

create table hist (
 h_c_id number
, h_c_d_id number
, h_c_w_id number
, h_d_id number
, h_w_id number
, h_date date
, h_amount number
, h_data varchar2(24)
)
 pctfree 5  initrans 4
 storage ( buffer_pool default )
 tablespace hist_0 ;
   set echo off
   spool off
   exit sql.sqlcode;
```

createtable_stoksql

```
set timing on
   set sqlblanklineson
   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 6500000000
hash is ( (s_i_id * 65000 + s_w_id) )
size 350
pctfree 0  initrans 3
storage ( buffer_pool keep )
tablespace stok_0;

create table stok (
s_i_id number
, s_w_id number
, s_quantity number
, s_ytd number
, s_order_cnt number
, s_remote_cnt number
, s_data varchar2(50)
, s_dist_01 char(24)
, s_dist_02 char(24)
, s_dist_03 char(24)
, s_dist_04 char(24)
, s_dist_05 char(24)
, s_dist_06 char(24)
, s_dist_07 char(24)
, s_dist_08 char(24)
, s_dist_09 char(24)
, s_dist_10 char(24)
)
cluster stokcluster (
s_i_id
, s_w_id
);
  set echo off
  spool off

  exit sql.sqlcode;
```

createtable_itemsql

```
set timing on
  set sqlblanklineson
  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 default )
tablespace item_0;

create table item (
i_id number(6,0)
, i_name varchar2(24)
, i_price number
, i_data varchar2(50)
, i_im_id number
)
cluster itemcluster (
i_id
);
  set echo off
  spool off
  exit sql.sqlcode;
```

createtable_ordrsql

```
set timing on
  set sqlblanklineson
  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 650000
  hash is (o_w_id  * 10 + o_d_id )
  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_nordsql

```
set timing on
  set sqlblanklineson
  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 650000
  hash is (no_w_id * 10 + no_d_id )
  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;
```

createindex_iware.sql

```
set timing on
  set sqlblanklineson
  spool createindex_iware.log ;
  set echo on ;
  drop index iware ;
    create unique index iware on ware ( w_id )
pctfree 1  initrans 3
storage ( buffer_pool default )
parallel 64
tablespace iware_0 ;
  set echo off
  spool off
  exit sql.sqlcode;
```

createindex_icust1sql

```
set timing on
  set sqlblanklineson
  spool createindex_icust1.log;
  set echo on ;
  drop index icust1 ;
    create unique index icust1 on cust ( c_w_id
, c_d_id
, c_id )
 pctfree 1  initrans 3
 storage ( buffer_pool default )
 parallel 16
 tablespace icust1_0 ;
  set echo off
  spool off
  exit sql.sqlcode;
```

createindex_icust2.sql

```
set timing on
  set sqlblanklineson
  spool createindex_icust2.log;
  set echo on ;
  drop index icust2 ;
    create unique index icust2 on cust ( c_last
, c_w_id
, c_d_id
, c_first
, c_id )
 pctfree 1  initrans 3
 storage ( buffer_pool default )
 parallel 16
 tablespace icust2_0 ;
  set echo off
  spool off
  exit sql.sqlcode;
```

createindex_idist.sql

```
set timing on
  set sqlblanklineson
  spool createindex_idist.log;
  set echo on ;
  drop index idist ;
    create unique index idist on dist ( d_w_id
, d_id )
 pctfree 5  initrans 3
 storage ( buffer_pool default )
 parallel 64
 tablespace idist_0 ;
  set echo off
  spool off
  exit sql.sqlcode;
```

createindex_istok.sql

```
set timing on
  set sqlblanklineson
  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 32
 tablespace istok_0 ;
  set echo off
  spool off
  exit sql.sqlcode;
```

createindex_iitem.sql

```
set timing on
  set sqlblanklineson
  spool createindex_iitem.log;
  set echo on ;
  drop index iitem;
    create unique index iitem on item ( i_id )
 pctfree 5  initrans 4
 storage ( buffer_pool default )
 parallel 64
 tablespace iitem_0 ;
  set echo off
  spool off
  exit sql.sqlcode;
```

createindex_iordr2.sql

```
set timing on
  set sqlblanklineson
  spool createindex_iordr2.log;
  set echo on ;
  drop index iordr2 ;
    create unique index iordr2 on ordr ( o_c_id
, o_d_id
, o_w_id
, o_id )
 pctfree 25  initrans 4
 storage ( buffer_pool default )
 parallel 64
 tablespace iordr2_0 ;
  set echo off
  spool off
  exit sql.sqlcode;
```

analyze.sql

```
spool analyze.log;
set echo on;

connect system/manager
execute dbms_stats.GATHER_TABLE_STATS(OWNNAME=>'TPCC', -
                                      TABNAME=>'DIST', -
                                      PARTNAME=>NULL, -
                                      ESTIMATE_PERCENT=>1, -
                                      BLOCK_SAMPLE=>TRUE, -
                                      METHOD_OPT=>'FOR ALL COLUMNS SIZE
1', -
                                      DEGREE=>10, -
                                      GRANULARITY=>'DEFAULT', -
                                      CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS(OWNNAME=>'TPCC', -
                                      TABNAME=>'ITEM', -
                                      PARTNAME=>NULL, -
                                      ESTIMATE_PERCENT=>10, -
                                      BLOCK_SAMPLE=>TRUE, -
                                      METHOD_OPT=>'FOR ALL COLUMNS SIZE
1', -
                                      DEGREE=>1, -
                                      GRANULARITY=>'DEFAULT', -
                                      CASCADE=>TRUE);

execute dbms_stats.GATHER_TABLE_STATS(OWNNAME=>'TPCC', -
                                      TABNAME=>'WARE', -
                                      PARTNAME=>NULL, -
                                      ESTIMATE_PERCENT=>10, -
                                      BLOCK_SAMPLE=>TRUE, -
                                      METHOD_OPT=>'FOR ALL COLUMNS SIZE
1', -
                                      DEGREE=>10, -
                                      GRANULARITY=>'DEFAULT', -
                                      CASCADE=>TRUE);


set echo off;
spool off;

exit sql.sqlcode;
```

C.3 Data Generation Code

tpccload.c

```
#ifdef RCSID
static char *RCSid =
 "$Header: tpccload.c 7030100.1 96/05/13 16:20:36 plai Generic<base> $ Copyr (c) 1993 Oracle";
#endif /* RCSID */

/*=================================================================+
|       Copyright (c) 1994  Oracle Corp, Redwood Shores, CA       |
|             OPEN SYSTEMS PERFORMANCE GROUP                      |
|                All Rights Reserved                              |
+=================================================================+
| FILENAME
|   tpccload.c
| DESCRIPTION
|   Load or generate  TPC-C database tables.
|   Usage: tpccload -M <# of wares> [options]
|          options: -A  load all tables
|                   -w  load ware table
|                   -d  load  dist table
|                   -c  load  cust table
|                    -i  load item table
|                   -s  load  stok table (cluster around s_w_id)
|                   -S  load  stok table (cluster around s_i_id)
|                   -h  load  hist table
|                   -n  load new-order table
|                   -o  <oline file> load order and order-line table
|                   -b  <ware#>  beginning ware number
|                   -e  <ware#>  ending ware number
|                   -j  <item#>  beginning item number (with -S)
|                   -k  <item#>  ending item number (with -S)
|                   -g  generate rows to standard output
+=================================================================*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <sys/types.h>
#include "tpcc.h"

#ifdef ORA_NT
#undef boolean
#include <process.h>
#include "dpbcore.h"
# define gettime dpbtimef
# define getcpu  dpbcpu
#define lrand48() ((long)rand() <<15 | rand())
#ifdef __STDC__
#  define PROTO(args)    args
#else
#  define PROTO(args)    ()
#endif
#endif

#define DISTARR 10                      /* dist insert array size          */
#define CUSTARR 100                     /* cust insert array size          */
#define STOCARR 100                     /* stok insert array size          */
#define ITEMARR 100                     /* item insert array size          */
#define HISTARR 100             /* hist insert array size    */
#define ORDEARR 100              /* order insert array size      */
#define NEWOARR 100              /* new order insert array size  */

#define DISTFAC 10                      /* max. dist id */
#define CUSTFAC 3000                    /* max. cust id*/
#define STOCFAC 100000                  /* max. stok id*/
#define ITEMFAC 100000                  /* max. item id                    */
#define HISTFAC 30000           /* history / warehouse  */
#define ORDEFAC 3000             /* order / district     */
#define NEWOFAC 900              /* new order / district */

#define C       0               /* constant in non-uniform dist. eqt. */
#define CNUM1   1               /* first constant in non-uniform dist. eqt. */
#define CNUM2   2               /* second constant in non-uniform dist. eqt. */
#define CNUM3   3               /* third constant in non-uniform dist. eqt. */

#define SEED    2               /* seed for random functions */

#define NOT_SERIALIZABLE  8177  /* ORA-08177: transaction not serializable*/
#define SNAPSHOT_TOO_OLD  1555  /* ORA-01555: snapshot too old */
#define RECOVERR -10
#define IRRECERR -20

#define SQLTXTW "INSERT INTO ware (w_id, w_ytd, w_tax, w_name, w_street_1, w_street_2, w_city,
w_state, w_zip) VALUES (:w_id, 30000000, :w_tax, :w_name, :w_street_1, \
 :w_street_2, :w_city, :w_state, :w_zip)"

#define SQLTXTD "INSERT INTO dist (d_id, d_w_id, d_ytd, d_tax, d_next_o_id, d_name, d_street_1,
d_street_2, d_city, d_state, d_zip) VALUES (:d_id, :d_w_id,3000000, :d_tax, \
   3001, :d_name, :d_street_1, :d_street_2, :d_city, :d_state, :d_zip)"

#define SQLTXTC "INSERT INTO cust (C_ID, C_D_ID, C_W_ID, C_FIRST, C_MIDDLE, C_LAST,
C_STREET_1, C_STREET_2, C_CITY, C_STATE, C_ZIP, C_PHONE, C_SINCE, C_CREDIT,
C_CREDIT_LIM, C_DISCOUNT, C_BALANCE, C_YTD_PAYMENT, C_PAYMENT_CNT,
C_DELIVERY_CNT, C_DATA) VALUES (:c_id, :c_d_id, :c_w_id, \
 :c_first, 'OE', :c_last, :c_street_1, :c_street_2, :c_city, :c_state, \
 :c_zip, :c_phone, SYSDATE, :c_credit, 5000000, :c_discount, -1000, 1000, 1, \
 0, :c_data)"

#define SQLTXTH "INSERT INTO hist (h_c_id, h_c_d_id, h_c_w_id, h_d_id, h_w_id, h_date, h_amount,
h_data) VALUES (:h_c_id, :h_c_d_id, :h_c_w_id, \
  :h_d_id, :h_w_id, SYSDATE, 1000, :h_data)"

#define SQLTXTS "INSERT INTO stok (s_i_id, s_w_id, s_quantity,s_dist_01, s_dist_02, s_dist_03,
s_dist_04, s_dist_05 , s_dist_06, s_dist_07, s_dist_08, s_dist_09, s_dist_10, s_ytd, s_order_cnt,
s_remote_cnt, s_data) \
VALUES (:s_i_id, :s_w_id, :s_quantity, \
  :s_dist_01, :s_dist_02, :s_dist_03, :s_dist_04, :s_dist_05, :s_dist_06, \
  :s_dist_07, :s_dist_08, :s_dist_09, :s_dist_10, 0, 0, 0, :s_data)" \

#define SQLTXTI "INSERT INTO item (I_ID,I_IM_ID,I_NAME,I_PRICE,I_DATA) VALUES (:i_id,
:i_im_id, :i_name, :i_price, \
  :i_data)"

#define SQLTXTO1 "INSERT INTO ordr (O_ID,
O_D_ID,O_W_ID,O_C_ID,O_ENTRY_D,O_CARRIER_ID,O_OL_CNT,O_ALL_LOCAL) \
  VALUES (:o_id, :o_d_id, :o_w_id, :o_c_id, \
  SYSDATE, :o_carrier_id, :o_ol_cnt, 1)"

#define SQLTXTO2 "INSERT INTO ordr (O_ID,
O_D_ID,O_W_ID,O_C_ID,O_ENTRY_D,O_CARRIER_ID,O_OL_CNT,O_ALL_LOCAL) \
  VALUES (:o_id, :o_d_id, :o_w_id, :o_c_id, \
  SYSDATE, 11, :o_ol_cnt, 1)"

#define SQLTXTOL1 "INSERT INTO ordl (OL_O_ID, OL_D_ID, OL_W_ID, OL_NUMBER,
OL_DELIVERY_D, OL_I_ID, OL_SUPPLY_W_ID, OL_QUANTITY, OL_AMOUNT, OL_DIST_INFO)
\
  VALUES (:ol_o_id, :ol_d_id, \
  :ol_w_id, :ol_number, SYSDATE, :ol_i_id, :ol_supply_w_id, 5, 0, \
  :ol_dist_info)"

#define SQLTXTOL2 "INSERT INTO ordl (OL_O_ID, OL_D_ID, OL_W_ID, OL_NUMBER,
OL_DELIVERY_D, OL_I_ID, OL_SUPPLY_W_ID, OL_QUANTITY, OL_AMOUNT, OL_DIST_INFO)
\
  VALUES (:ol_o_id, :ol_d_id, \
  :ol_w_id, :ol_number, to_date('01-Jan-1811'), :ol_i_id, :ol_supply_w_id, 5, :ol_amount, \
  :ol_dist_info)"

#define SQLTXTNO "INSERT INTO nord (no_o_id, no_d_id, no_w_id) VALUES (:no_o_id, :no_d_id,
:no_w_id)"

#define SQLTXTENHA "alter session set \"_enable_hash_overflow\"=true"
#define SQLTXTDIHA "alter session set \"_enable_hash_overflow\"=false"

static char *lastname[] = {
  "BAR",
  "OUGHT",
  "ABLE",
  "PRI",
  "PRES",
  "ESE",
  "ANTI",
  "CALLY",
  "ATION",
  "EING"
};

char num9[10];
char num16[17];
char str2[3];
char str24[15][25];
int randperm3000[3000];

void initperm();
void randstr();
void randdatastr();
void randnum();
void randlastname (char*, int);
int NURand();
void sysdate();

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

OCIStmt *curw;
OCIStmt *curd;
OCIStmt *curc;
OCIStmt *curh;
OCIStmt *curs;
OCIStmt *curi;
OCIStmt *curo1;
OCIStmt *curo2;
OCIStmt *curol1;
OCIStmt *curol2;
OCIStmt *curno;

OCIBind *w_id_bp = (OCIBind *) 0;
OCIBind *w_name_bp = (OCIBind *) 0;
OCIBind *w_street1_bp = (OCIBind *) 0;
OCIBind *w_street2_bp = (OCIBind *) 0;
OCIBind *w_city_bp = (OCIBind *) 0;
OCIBind *w_state_bp = (OCIBind *) 0;
OCIBind *w_zip_bp = (OCIBind *) 0;
OCIBind *w_tax_bp = (OCIBind *) 0;
```

```
OCIBind *d_id_bp = (OCIBind *) 0;
OCIBind *d_w_id_bp = (OCIBind *) 0;
OCIBind *d_name_bp = (OCIBind *) 0;
OCIBind *d_street1_bp = (OCIBind *) 0;
OCIBind *d_street2_bp = (OCIBind *) 0;
OCIBind *d_city_bp = (OCIBind *) 0;
OCIBind *d_state_bp = (OCIBind *) 0;
OCIBind *d_zip_bp = (OCIBind *) 0;
OCIBind *d_tax_bp = (OCIBind *) 0;

OCIBind *c_id_bp = (OCIBind *) 0;
OCIBind *c_d_id_bp = (OCIBind *) 0;
OCIBind *c_w_id_bp = (OCIBind *) 0;
OCIBind *c_first_bp = (OCIBind *) 0;
OCIBind *c_last_bp = (OCIBind *) 0;
OCIBind *c_street1_bp = (OCIBind *) 0;
OCIBind *c_street2_bp = (OCIBind *) 0;
OCIBind *c_city_bp = (OCIBind *) 0;
OCIBind *c_state_bp = (OCIBind *) 0;
OCIBind *c_zip_bp = (OCIBind *) 0;
OCIBind *c_phone_bp = (OCIBind *) 0;
OCIBind *c_discount_bp = (OCIBind *) 0;
OCIBind *c_credit_bp = (OCIBind *) 0;
OCIBind *c_data_bp = (OCIBind *) 0;

OCIBind *i_id_bp = (OCIBind *) 0;
OCIBind *i_im_id_bp = (OCIBind *) 0;
OCIBind *i_name_bp = (OCIBind *) 0;
OCIBind *i_price_bp = (OCIBind *) 0;
OCIBind *i_data_bp = (OCIBind *) 0;

OCIBind *s_i_id_bp = (OCIBind *) 0;
OCIBind *s_w_id_bp = (OCIBind *) 0;
OCIBind *s_quantity_bp = (OCIBind *) 0;
OCIBind *s_dist_01_bp = (OCIBind *) 0;
OCIBind *s_dist_02_bp = (OCIBind *) 0;
OCIBind *s_dist_03_bp = (OCIBind *) 0;
OCIBind *s_dist_04_bp = (OCIBind *) 0;
OCIBind *s_dist_05_bp = (OCIBind *) 0;
OCIBind *s_dist_06_bp = (OCIBind *) 0;
OCIBind *s_dist_07_bp = (OCIBind *) 0;
OCIBind *s_dist_08_bp = (OCIBind *) 0;
OCIBind *s_dist_09_bp = (OCIBind *) 0;
OCIBind *s_dist_10_bp = (OCIBind *) 0;
OCIBind *s_data_bp = (OCIBind *) 0;

OCIBind *h_c_id_bp = (OCIBind *) 0;
OCIBind *h_c_d_id_bp = (OCIBind *) 0;
OCIBind *h_c_w_id_bp = (OCIBind *) 0;
OCIBind *h_d_id_bp = (OCIBind *) 0;
OCIBind *h_w_id_bp = (OCIBind *) 0;
OCIBind *h_data_bp = (OCIBind *) 0;

OCIBind *ol_o_id_bp = (OCIBind *) 0;
OCIBind *ol_d_id_bp = (OCIBind *) 0;
OCIBind *ol_w_id_bp = (OCIBind *) 0;
OCIBind *ol_i_id_bp = (OCIBind *) 0;
OCIBind *ol_number_bp = (OCIBind *) 0;
OCIBind *ol_supply_w_id_bp = (OCIBind *) 0;
OCIBind *ol_dist_info_bp = (OCIBind *) 0;
OCIBind *ol_amount_bp = (OCIBind *) 0;

OCIBind *o_id_bp = (OCIBind *) 0;
OCIBind *o_d_id_bp = (OCIBind *) 0;
OCIBind *o_w_id_bp = (OCIBind *) 0;
OCIBind *o_c_id_bp = (OCIBind *) 0;
OCIBind *o_carrier_id_bp = (OCIBind *) 0;
OCIBind *o_ol_cnt_bp  = (OCIBind *) 0;
OCIBind *o_ocnt_bp = (OCIBind *) 0;
OCIBind *o_olcnt_bp  = (OCIBind *) 0;


OCIBind *no_o_id_bp  = (OCIBind *) 0;
OCIBind *no_d_id_bp  = (OCIBind *) 0;
OCIBind *no_w_id_bp  = (OCIBind *) 0;

void myusage()
{
  fprintf (stderr, "\n");
  fprintf (stderr, "Usage:\ttpccload -M <multiplier> [options]\n");
  fprintf (stderr, "options:\n");
  fprintf (stderr, "\t-A :\tload all tables\n");
  fprintf (stderr, "\t-w :\tload ware table\n");
  fprintf (stderr, "\t-d :\tload dist table\n");
  fprintf (stderr, "\t-c :\tload cust table\n");
  fprintf (stderr, "\t-i :\tload item table\n");
  fprintf (stderr, "\t-s :\tload stok table (cluster around s_w_id)\n");
  fprintf (stderr, "\t-S :\tload stok table (cluster around s_i_id)\n");
  fprintf (stderr, "\t-h :\tload hist table\n");
  fprintf (stderr, "\t-n :\tload new-order table\n");
  fprintf (stderr, "\t-o <oline file> :\tload order and order-line table\n");
  fprintf (stderr, "\t-b <ware#> :\tbeginning ware number\n");
  fprintf (stderr, "\t-e <ware#> :\tending ware number\n");
  fprintf (stderr, "\t-j <item#> :\tbeginning item number (with -S)\n");
  fprintf (stderr, "\t-k <item#> :\tending item number (with -S)\n");
  fprintf (stderr, "\t-g :\tgenerate rows to standard output\n");
  fprintf (stderr,"\t   $tpcc_bench must be set to the location of the kit\n");
  fprintf (stderr, "\n");
  exit(1);
}

int sqlfile(fnam,linebuf)
char   *fnam;
text   *linebuf;
{
   FILE *fd;
   int nulpt = 0;
   char realfile[512];

   sprintf(realfile,"%s",fnam);
   fd = fopen(realfile,"r");
   if (!fd)
   {
     return (0);
   }
   while (fgets((char *)linebuf+nulpt,SQL_BUF_SIZE, fd))
   {
      nulpt = strlen((char *)linebuf);
   }
   return(nulpt);
}

void quit()
{
  OCIERROR(errhp,OCISessionEnd ( tpcsvc,errhp, tpcusr, OCI_DEFAULT));
  OCIERROR(errhp,OCIServerDetach ( tpcsrv, errhp, OCI_DEFAULT));
  OCIHandleFree((dvoid *)tpcusr, OCI_HTYPE_SESSION);
  OCIHandleFree((dvoid *)tpcsvc, OCI_HTYPE_SVCCTX);
  OCIHandleFree((dvoid *)errhp, OCI_HTYPE_ERROR);
  OCIHandleFree((dvoid *)tpcsrv, OCI_HTYPE_SERVER);
  OCIHandleFree((dvoid *)tpcenv, OCI_HTYPE_ENV);
}

void main(argc, argv)
int argc;
char *argv[];
{
  char *uid="tpcc";
  char *pwd="tpcc";
  int scale=0;
  int i, j;
  int loop;
  int loopcount;
  int cid;
  int dwid;
  int cdid;
  int cwid;
  int sid;
  int swid;
  int olcnt;
  int nrows;
  int row;

  int w_id;
  char w_name[11];
  char w_street_1[21];
  char w_street_2[21];
  char w_city[21];
  char w_state[2];
  char w_zip[9];
  float w_tax;

  int d_id[10];
  int d_w_id[10];
  char d_name[10][11];
  char d_street_1[10][21];
  char d_street_2[10][21];
  char d_city[10][21];
  char d_state[10][2];
  char d_zip[10][9];
  float d_tax[10];

  int c_id[100];
  int c_d_id[100];
  int c_w_id[100];
  char c_first[100][17];
  char c_last[100][17];
  char c_street_1[100][21];
  char c_street_2[100][21];
  char c_city[100][21];
  char c_state[100][2];
  char c_zip[100][9];
  char c_phone[100][16];
  char c_credit[100][2];
  float c_discount[100];
  char c_data[100][501];

  int i_id[100];
  int i_im_id[100];
  int i_price[100];
  char i_name[100][25];
  char i_data[100][51];

  int s_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:e:j:k:g";
  int do_A=0;
  int do_w=0;
  int do_d=0;
  int do_i=0;
  int do_c=0;
  int do_s=0;
  int do_S=0;
  int do_h=0;
  int do_o=0;
  int do_n=0;
  int gen=0;
  int bware=1;
  int eware=0;
  int bitem=1;
  int eitem=0;

  FILE *olfp=NULL;
  char olfname[100];
  char* basename;
  int status;
#ifdef ORA_NT
  char fname[100];
  FILE *logfile;
#endif /* ORA_NT */

/*---------------------------------------------------------------+
| Parse command line -- look for scale factor.                   |
+---------------------------------------------------------------*/

  if (argc == 1) {
    myusage();
  }

#ifdef ORA_NT
  end_args = argv + argc;
  for (++argv; argv < end_args; )
  {
    arg_ptr = *argv++;

      if (*arg_ptr != '-')
      {
        myusage();
      } else
      {
     switch (arg_ptr[1]) {
      case '?': myusage();
              break;
      case 'M': scale = atoi (*argv++);
              break;
      case 'A': do_A = 1;
              break;
      case 'w': do_w = 1;
              break;
      case 'd': do_d = 1;
              break;
      case 'c': do_c = 1;
              break;
      case 'i': do_i = 1;
              break;
      case 's': do_s = 1;
              break;
      case 'S': do_S = 1;
              break;
      case 'h': do_h = 1;
              break;
      case 'n': do_n = 1;
              break;
      case 'o': do_o = 1;
               strcpy (olfname, *argv++);
              break;
      case 'b': bware = atoi (*argv++);
              break;
      case 'e': eware = atoi (*argv++);
              break;
      case 'j': bitem = atoi (*argv++);
              break;
      case 'k': eitem = atoi (*argv++);
              break;
      case 'g': gen = 1;
               strcpy (fname, *argv++);
              break;
      case 'l': logfile=fopen(*argv++,"w");
              break;
      default: fprintf (stderr, "THIS SHOULD NEVER HAPPEN!!!\n");
              fprintf (stderr, "(reached default case in getopt ())\n");
              myusage();
     }
      }
  }


#else

  while ((opt = getopt (argc, argv, argstr)) != -1) {
    switch (opt) {
      case '?': myusage();
              break;
      case 'M': scale = atoi (optarg);
              break;
      case 'A': do_A = 1;
              break;
      case 'w': do_w = 1;
              break;
      case 'd': do_d = 1;
              break;
      case 'c': do_c = 1;
              break;
      case 'i': do_i = 1;
```

```
            break;
        case 's': do_s = 1;
            break;
        case 'S': do_S = 1;
            break;
        case 'h': do_h = 1;
            break;
        case 'n': do_n = 1;
            break;
        case 'o': do_o = 1;
            strcpy (olfname, optarg);
            break;
        case 'b': bware = atoi (optarg);
            break;
        case 'e': eware = atoi (optarg);
            break;
        case 'j': bitem = atoi (optarg);
            break;
        case 'k': eitem = atoi (optarg);
            break;
        case 'g': gen = 1;
            break;
        default: fprintf (stderr, "THIS SHOULD NEVER HAPPEN!!!\n");
            fprintf (stderr, "(reached default case in getopt ())\n");
            myusage ();
    }
  }

# endif /* ORA_NT */

/*----------------------------------------------------------------*|
|                  Rudimentary error checking                    |
|*----------------------------------------------------------------*/

  if (scale  < 1) {
    fprintf (stderr, "Invalid scale factor:'%d'\n", scale);
    myusage ();
  }

  if (!(do_A || do_w || do_d || do_c || do_i || do_s || do_S || do_h || do_o ||
        do_n)) {
    fprintf (stderr, "What should I load???\n");
    myusage ();
  }

  if (gen && (do_A || (do_w + do_d + do_c + do_i + do_s + do_S + do_h + do_o +
                       do_n > 1))) {
    fprintf (stderr, "Can only generate table one at a time\n");
    myusage ();
  }

  if (do_S && (do_A || do_s)) {
    fprintf (stderr, "Cluster stock table around s_w_id or s_i_id?\n");
    myusage ();
  }

  if (eware <= 0)
    eware = scale;
  if (eitem <= 0)
    eitem = STOCFAC;

  if (do_S) {
    if ((bitem  < 1) || (bitem > STOCFAC)) {
      fprintf (stderr, "Invalid beginning item number:%d'\n", bitem);
      myusage ();
    }

    if ((eitem  < bitem) || (eitem > STOCFAC)) {
      fprintf (stderr, "Invalid ending item number:'%d'\n", eitem);
      myusage ();
    }
  }
  if (do_o) {
    if ((basename = getenv ("tpcc_bench")) == NULL)
    {
      fprintf (stderr, "$tpcc_bench is not set");
      myusage ();
    }
  }

  if ((bware  < 1) || (bware > scale)) {
    fprintf (stderr, "Invalid beginning warehouse number:'%d'\n", bware);
    myusage ();
  }

  if ((eware  < bware) || (eware > scale)) {
    fprintf (stderr, "Invalid ending warehouse number:'%d'\n", eware);
    myusage ();
  }

  if (gen && do_o) {
    if ((olfp = fopen (olfname, "w")) == NULL) {
      fprintf (stderr, "Can't open '%s' for writing order lines\n", olfname);
      myusage ();
    }


  }
```

```
/*----------------------------------------------------------------+
| Prepare to insert into database.                                |
+----------------------------------------------------------------*/

  sysdate (sdate);
  if (!gen) {

    /* log on to Oracle */

    OCIInitialize(OCI_DEFAULT|OCI_OBJECT,(dvoid*)0,0,0,0);
    OCIEnvInit(&tpcenv, OCI_DEFAULT, 0, (dvoid **)0);
    OCIHandleAlloc((dvoid*)tpcenv, (dvoid **)&tpcsrv, OCI_HTYPE_SERVER, 0 , (dvoid **)0);
    OCIHandleAlloc((dvoid*)tpcenv, (dvoid **)&errhp, OCI_HTYPE_ERROR, 0 , (dvoid **)0);
    OCIHandleAlloc((dvoid*)tpcenv, (dvoid **)&tpcsvc, OCI_HTYPE_SVCCTX, 0 , (dvoid **)0);
    OCIServerAttach(tpcsrv, errhp, (text *)0,0,OCI_DEFAULT);
    OCIAttrSet((dvoid *)tpcsvc, OCI_HTYPE_SVCCTX, (dvoid *)tpcsrv,
            (ub4)0,OCI_ATTR_SERVER, errhp);
    OCIHandleAlloc((dvoid*)tpcenv, (dvoid **)&tpcusr, OCI_HTYPE_SESSION, 0 , (dvoid **)0);
    OCIAttrSet((dvoid *)tpcusr, OCI_HTYPE_SESSION, (dvoid *)uid,
            (ub4)strlen(uid),OCI_ATTR_USERNAME, errhp);
    OCIAttrSet((dvoid *)tpcusr, OCI_HTYPE_SESSION, (dvoid *)pwd, (ub4)strlen(pwd),
          OCI_ATTR_PASSWORD, errhp);
    OCIERROR(errhp, OCISessionBegin(tpcsvc, errhp, tpcusr, OCI_CRED_RDBMS, OCI_DEFAULT));

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

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

    /* open cursors and parse statement */
    if (do_A || do_w)  {
      OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid**)(&curw), OCI_HTYPE_STMT, 0,
(dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curw, errhp, (text *)SQLTXTW,
            strlen((char *)SQLTXTW), (ub4) OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

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

    if (do_A || do_c) {
      OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid**)(&curc), OCI_HTYPE_STMT, 0, (dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curc, errhp, (text *)SQLTXTC,
            strlen((char *)SQLTXTC), (ub4) OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

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

    if (do_A || do_s || do_S) {
      OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid**)(&curs), OCI_HTYPE_STMT, 0, (dvoid**)0));
      OCIERROR(errhp,OCIStmtPrepare(curs, errhp, (text *)SQLTXTS,
            strlen((char *)SQLTXTS), (ub4) OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

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

    if (do_A || do_o) {
      int stat;
      char fname[160];
      OCIERROR(errhp,OCIHandleAlloc(tpcenv,(dvoid**)(&curo1), OCI_HTYPE_STMT, 0,
(dvoid**)0));
      DISCARD strcpy(fname,basename);
      DISCARD strcat(fname, "/");
      DISCARD strcat(fname, "benchrun/blocks/load_ordordl.sql");
      stat = sqlfile(fname,stmbuf);
      if (!stat)
      {
        fprintf(stderr, "unable to open %s \n",fname);
        quit();
        exit(1);
      }
      OCIERROR(errhp,OCIStmtPrepare(curo1, errhp, stmbuf,
            strlen((char *)stmbuf), (ub4) OCI_NTV_SYNTAX, (ub4) OCI_DEFAULT));
    }

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

    /* bind variables*/

    /* warehouse */

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

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

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

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

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

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

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

    OCIERROR(errhp, OCIBindByName(curw,&w_zip_bp, errhp, (text *)":w_zip",
                        strlen(":w_zip"), (ub1 *)w_zip, 9, SQLT_CHR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

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

  /* district */

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

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

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

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

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

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

    OCIERROR(errhp, OCIBindByName(curd,&d_state_bp, errhp, (text *)":d_state",
            strlen(":d_state"), (ub1 *)d_state, 2, SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curd,&d_zip_bp, errhp, (text *)":d_zip",
            strlen(":d_zip"), (ub1 *)d_zip, 9, SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curd,&d_tax_bp, errhp, (text *)":d_tax",
            strlen(":d_tax"), (ub1 *)d_tax, sizeof(float), SQLT_FLT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
  }

  /* customer */

  if (do_A || do_c) {
    OCIERROR(errhp, OCIBindByName(curc,&c_id_bp, errhp, (text *)":c_id",
            strlen(":c_id"), (ub1 *)c_id, sizeof(int), SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,&c_d_id_bp, errhp, (text *)":c_d_id",
            strlen(":c_d_id"), (ub1 *)c_d_id, sizeof(int), SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,&c_w_id_bp, errhp, (text *)":c_w_id",
            strlen(":c_w_id"), (ub1 *)c_w_id, sizeof(int), SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,&c_first_bp, errhp, (text *)":c_first",
                        strlen(":c_first"), (ub1 *)c_first, 17, SQLT_STR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,&c_last_bp, errhp, (text *)":c_last",
                        strlen(":c_last"), (ub1 *)c_last, 17, SQLT_STR,
                        (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,&c_street1_bp, errhp, (text *)":c_street_1",
            strlen(":c_street_1"), (ub1 *)c_street_1, 21, SQLT_STR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,&c_street2_bp, errhp, (text *)":c_street_2",
            strlen(":c_street_2"), (ub1 *)c_street_2, 21, SQLT_STR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,&c_city_bp, errhp, (text *)":c_city",
            strlen(":c_city"), (ub1 *)c_city, 21, SQLT_STR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,&c_state_bp, errhp, (text *)":c_state",
            strlen(":c_state"), (ub1 *)c_state, 2, SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,&c_zip_bp, errhp, (text *)":c_zip",
            strlen(":c_zip"), (ub1 *)c_zip, 9, SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,&c_phone_bp, errhp, (text *)":c_phone",
            strlen(":c_phone"), (ub1 *)c_phone, 16, SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,&c_credit_bp, errhp, (text *)":c_credit",
            strlen(":c_credit"), (ub1 *)c_credit, 2, SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,&c_discount_bp, errhp, (text *)":c_discount",
            strlen(":c_discount"), (ub1 *)c_discount, sizeof(float), SQLT_FLT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curc,&c_data_bp, errhp, (text *)":c_data",
            strlen(":c_data"), (ub1 *)c_data, 501, SQLT_STR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
  }

  /* item */

  if (do_A || do_i) {
    OCIERROR(errhp, OCIBindByName(curi,&i_id_bp, errhp, (text *)":i_id",
            strlen(":i_id"), (ub1 *)i_id, sizeof(int), SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curi,&i_im_id_bp,errhp, (text *)":i_im_id",
            strlen(":i_im_id"),(ub1 *)i_im_id, sizeof(int), SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curi,&i_name_bp,errhp, (text *)":i_name",
            strlen(":i_name"),(ub1 *)i_name, 25, SQLT_STR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curi,&i_price_bp, errhp, (text *)":i_price",
            strlen(":i_price"), (ub1 *)i_price, sizeof(int), SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curi,&i_data_bp, errhp, (text *)":i_data",
            strlen(":i_data"), (ub1 *)i_data, 51, SQLT_STR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
  }

  /* stock */

  if (do_A || do_s || do_S) {
    OCIERROR(errhp, OCIBindByName(curs,&s_i_id_bp, errhp, (text *)":s_i_id",
            strlen(":s_i_id"), (ub1 *)s_i_id, sizeof(int), SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
```

```
    OCIERROR(errhp, OCIBindByName(curs,&s_w_id_bp,errhp, (text *)":s_w_id",
            strlen(":s_w_id"),(ub1 *)s_w_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curs,&s_quantity_bp,errhp, (text *)":s_quantity",
            strlen(":s_quantity"),(ub1 *)s_quantity, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curs,&s_dist_01_bp,errhp, (text *)":s_dist_01",
            strlen(":s_dist_01"),(ub1 *)s_dist_01, 24,SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


    OCIERROR(errhp, OCIBindByName(curs,&s_dist_02_bp,errhp, (text *)":s_dist_02",
            strlen(":s_dist_02"),(ub1 *)s_dist_02, 24,SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


    OCIERROR(errhp, OCIBindByName(curs,&s_dist_03_bp,errhp, (text *)":s_dist_03",
            strlen(":s_dist_03"),(ub1 *)s_dist_03, 24,SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


    OCIERROR(errhp, OCIBindByName(curs,&s_dist_04_bp,errhp, (text *)":s_dist_04",
            strlen(":s_dist_04"),(ub1 *)s_dist_04, 24,SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


    OCIERROR(errhp, OCIBindByName(curs,&s_dist_05_bp,errhp, (text *)":s_dist_05",
            strlen(":s_dist_05"),(ub1 *)s_dist_05, 24,SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


    OCIERROR(errhp, OCIBindByName(curs,&s_dist_06_bp,errhp, (text *)":s_dist_06",
            strlen(":s_dist_06"),(ub1 *)s_dist_06, 24,SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


    OCIERROR(errhp, OCIBindByName(curs,&s_dist_07_bp,errhp, (text *)":s_dist_07",
            strlen(":s_dist_07"),(ub1 *)s_dist_07, 24,SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


    OCIERROR(errhp, OCIBindByName(curs,&s_dist_08_bp,errhp, (text *)":s_dist_08",
            strlen(":s_dist_08"),(ub1 *)s_dist_08, 24,SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


    OCIERROR(errhp, OCIBindByName(curs,&s_dist_09_bp,errhp, (text *)":s_dist_09",
            strlen(":s_dist_09"),(ub1 *)s_dist_09, 24,SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));


    OCIERROR(errhp, OCIBindByName(curs,&s_dist_10_bp,errhp, (text *)":s_dist_10",
            strlen(":s_dist_10"),(ub1 *)s_dist_10, 24,SQLT_CHR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curs,&s_data_bp,errhp, (text *)":s_data",
            strlen(":s_data"),(ub1 *)s_data, 51,SQLT_STR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
}

/* history */

if (do_A || do_h) {
  OCIERROR(errhp, OCIBindByName(curh,&h_c_id_bp,errhp, (text *)":h_c_id",
            strlen(":h_c_id"),(ub1 *)h_c_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curh,&h_c_d_id_bp,errhp, (text *)":h_c_d_id",
            strlen(":h_c_d_id"),(ub1 *)h_d_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curh,&h_c_w_id_bp,errhp, (text *)":h_c_w_id",
            strlen(":h_c_w_id"),(ub1 *)h_w_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curh,&h_d_id_bp,errhp, (text *)":h_d_id",
            strlen(":h_d_id"),(ub1 *)h_d_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curh,&h_w_id_bp,errhp, (text *)":h_w_id",
            strlen(":h_w_id"),(ub1 *)h_w_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curh,&h_data_bp,errhp, (text *)":h_data",
            strlen(":h_data"),(ub1 *)h_data, 25,SQLT_STR,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

}

/* order and order_line (delivered) */

if (do_A || do_o) {

  for (i = 0; i < ORDEARR; i++ ) {
    o_id_len[i] = sizeof(int);
    o_d_id_len[i] = sizeof(int);
    o_w_id_len[i] = sizeof(int);
    o_c_id_len[i] = sizeof(int);
    o_carrier_id_len[i] = sizeof(int);
    o_ol_cnt_len[i] = sizeof(int);
  }

  OCIERROR(errhp, OCIBindByName(curo1,&ol_o_id_bp,errhp, (text *)":ol_o_id",
            strlen(":ol_o_id"),(ub1 *)ol_o_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)ol_o_id_len, (ub2 *)0,
            (ub4) 15*ORDEARR, (ub4 *)&ol_o_id_clen, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&ol_d_id_bp,errhp, (text *)":ol_d_id",
            strlen(":ol_d_id"),(ub1 *)ol_d_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)ol_d_id_len, (ub2 *)0,
            (ub4) 15*ORDEARR, (ub4 *) &ol_d_id_clen, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&ol_w_id_bp,errhp, (text *)":ol_w_id",
            strlen(":ol_w_id"),(ub1 *)ol_w_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)ol_w_id_len, (ub2 *)0,
            (ub4) 15*ORDEARR, (ub4 *) &ol_w_id_clen, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&ol_number_bp,errhp, (text *)":ol_number",
            strlen(":ol_number"),(ub1 *)ol_number, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)ol_number_len, (ub2 *)0,
            (ub4) 15*ORDEARR, (ub4 *) &ol_number_clen, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&ol_i_id_bp,errhp, (text *)":ol_i_id",
            strlen(":ol_i_id"),(ub1 *)ol_i_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)ol_i_id_len, (ub2 *)0,
            (ub4) 15*ORDEARR, (ub4 *) &ol_i_id_clen, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&ol_supply_w_id_bp,errhp, (text *)":ol_supply_w_id",
            strlen(":ol_supply_w_id"),(ub1 *)ol_supply_w_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)ol_supply_w_id_len,(ub2 *)0,
            (ub4) 15*ORDEARR, (ub4 *) &ol_supply_w_id_clen,(ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&ol_dist_info_bp,errhp, (text *)":ol_dist_info",
            strlen(":ol_dist_info"),(ub1 *)ol_dist_info, 24,SQLT_CHR,
            (dvoid *) 0, (ub2 *)ol_dist_info_len,(ub2 *)0,
            (ub4) 15*ORDEARR, (ub4 *) &ol_dist_info_clen,(ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&ol_amount_bp,errhp, (text *)":ol_amount",
            strlen(":ol_amount"),(ub1 *)ol_amount, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)ol_amount_len, (ub2 *)0,
            (ub4) 15*ORDEARR, (ub4 *) &ol_amount_clen,(ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&o_id_bp,errhp, (text *)":o_id",
            strlen(":o_id"),(ub1 *)o_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)o_id_len, (ub2 *)0,
            (ub4) ORDEARR, (ub4 *) &o_id_clen, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&o_d_id_bp,errhp, (text *)":o_d_id",
            strlen(":o_d_id"),(ub1 *)o_d_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)o_d_id_len, (ub2 *)0,
            (ub4) ORDEARR, (ub4 *) &o_d_id_clen, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&o_w_id_bp,errhp, (text *)":o_w_id",
            strlen(":o_w_id"),(ub1 *)o_w_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)o_w_id_len, (ub2 *)0,
            (ub4) ORDEARR, (ub4 *) &o_w_id_clen, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&o_c_id_bp,errhp, (text *)":o_c_id",
            strlen(":o_c_id"),(ub1 *)o_c_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)o_c_id_len, (ub2 *)0,
            (ub4) ORDEARR, (ub4 *) &o_c_id_clen, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&o_carrier_id_bp,errhp, (text *)":o_carrier_id",
            strlen(":o_carrier_id"),(ub1 *)o_carrier_id, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)o_carrier_id_len, (ub2 *)0,
            (ub4) ORDEARR, (ub4 *) &o_carrier_id_clen, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&o_ol_cnt_bp,errhp, (text *)":o_ol_cnt",
            strlen(":o_ol_cnt"),(ub1 *)o_ol_cnt, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)o_ol_cnt_len, (ub2 *)0,
            (ub4) ORDEARR, (ub4 *) &o_ol_cnt_clen, (ub4) OCI_DEFAULT));

  OCIERROR(errhp, OCIBindByName(curo1,&o_ocnt_bp,errhp, (text *)":order_rows",
            strlen(":order_rows"),(ub1 *)&o_cnt, sizeof(int),SQLT_INT,
            (dvoid *) 0, (ub2 *)0, (ub2 *)0,
            (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
```

```
    OCIERROR(errhp, OCIBindByName(curo1, &o_olcnt_bp, errhp, (text *)":ordl_rows",
                strlen(":ordl_rows"), (ub1 *)&ol_cnt, sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
  }

  /* new order */

  if (do_A || do_n) {
    OCIERROR(errhp, OCIBindByName(curno, &no_o_id_bp, errhp, (text *)":no_o_id",
                strlen(":no_o_id"), (ub1 *)no_o_id, sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curno, &no_d_id_bp, errhp, (text *)":no_d_id",
                strlen(":no_d_id"), (ub1 *)no_d_id, sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));

    OCIERROR(errhp, OCIBindByName(curno, &no_w_id_bp, errhp, (text *)":no_w_id",
                strlen(":no_w_id"), (ub1 *)no_w_id, sizeof(int), SQLT_INT,
                (dvoid *) 0, (ub2 *)0, (ub2 *)0,
                (ub4) 0, (ub4 *) 0, (ub4) OCI_DEFAULT));
  }
 }

/*---------------------------------------------------------------+
| Initialize random number generator                             |
+---------------------------------------------------------------*/

 srand (SEED);
#ifndef ORA_NT
 srand48 (SEED);
#endif
 initperm ();


/*---------------------------------------------------------------+
| Load the WAREHOUSE table.                                      |
+---------------------------------------------------------------*/

 if (do_A || do_w) {
   nrows = eware - bware + 1;

   fprintf (stderr, "Loading/generating warehouse: w%d - w%d (%d rows)\n",
            bware, eware, nrows);

   begin_time = gettime ();
   begin_cpu = getcpu ();

   for (loop = bware; loop <= eware; loop++) {

     w_tax = (float) ((lrand48 () % 2001) * 0.0001);
     randstr (w_name, 6, 10);
     randstr (w_street_1, 10, 20);
     randstr (w_street_2, 10, 20);
     randstr (w_city, 10, 20);
     randstr (str2, 2, 2);
     randnum (num9, 9);
     num9[4] = num9[5] = num9[6] = num9[7] = num9[8] = '1';

     if (gen) {
       printf ("%d 30000000 %6.4f %s %s %s %s %s %s\n", loop, w_tax,
               w_name, w_street_1, w_street_2, w_city, str2, num9);
       fflush (stdout);
     }
     else {
       w_id = loop;
       strncpy (w_state, str2, 2);
       strncpy (w_zip, num9, 9);

                   status = OCIStmtExecute(tpcsvc, curw, errhp, (ub4) 1, (ub4) 0,
                 (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                 (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
                   if (status != OCI_SUCCESS) {
                     fprintf (stderr, "Error at ware %d\n", loop);
                     OCIERROR(errhp, status);
                     quit ();
         exit (1);
       }
     }
   }

   end_time = gettime ();
   end_cpu = getcpu ();
   fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f cpu)\n\n",
            nrows, end_time - begin_time, end_cpu - begin_cpu);
 }

/*---------------------------------------------------------------+
| Load the DISTRICT table.                                       |
+---------------------------------------------------------------*/

 if (do_A || do_d) {
   nrows = (eware - bware + 1) * DISTFAC;

   fprintf (stderr, "Loading/generating district: w%d - w%d (%d rows)\n",
            bware, eware, nrows);

   begin_time = gettime ();
   begin_cpu = getcpu ();

   dwid = bware - 1;

   for (row = 0; row < nrows; ) {
     dwid++;

     for (i = 0; i < DISTARR; i++, row++) {
       d_tax[i] = (float) ((lrand48 () % 2001) * 0.0001);
       randstr (d_name[i], 6, 10);
       randstr (d_street_1[i], 10, 20);
       randstr (d_street_2[i], 10, 20);
       randstr (d_city[i], 10, 20);
       randstr (str2, 2, 2);
       randnum (num9, 9);
       num9[4] = num9[5] = num9[6] = num9[7] = num9[8] = '1';

       if (gen) {
         printf ("%d %d 3000000 %6.4f 3001 %s %s %s %s %s %s\n",
                 i + 1, dwid, d_tax[i], d_name[i], d_street_1[i],
                                        d_street_2[i], d_city[i], str2, num9 );
       }
       else {
         d_id[i] = i + 1;
         d_w_id[i] = dwid;
         strncpy (d_state[i], str2, 2);
         strncpy (d_zip[i], num9, 9);
       }
     }

     if (gen) {
       fflush (stdout);
     }
     else {
       status = OCIStmtExecute(tpcsvc, curd, errhp, (ub4) DISTARR, (ub4) 0,
                 (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                 (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
       if (status != OCI_SUCCESS) {
                                       fprintf (stderr, "Aborted at ware %d, dist 1\n", dwid);
           OCIERROR(errhp, status);
           quit ();
           exit (1);
       }
     }
   }

   end_time = gettime ();
   end_cpu = getcpu ();
   fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f cpu)\n\n",
            nrows, end_time - begin_time, end_cpu - begin_cpu);
 }

/*---------------------------------------------------------------+
| Load the CUSTOMER table.                                       |
+---------------------------------------------------------------*/

 if (do_A || do_c) {

   nrows = (eware - bware + 1) * CUSTFAC * DISTFAC;

   fprintf (stderr, "Loading/generating customer: w%d - w%d (%d rows)\n   ",
            bware, eware, nrows);

   if (getenv("tpcc_hash_overflow")) {
     fprintf(stderr,"Hash overflow is enabled\n");
     OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);
     sprintf ((char *) stmbuf, SQLTXTENHA);
     OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf),
                    OCI_NTV_SYNTAX, OCI_DEFAULT);
     OCIERROR(errhp,OCIStmtExecute(tpcsvc, curi, errhp,1,0,0,0,OCI_DEFAULT));
     OCIHandleFree(curi, OCI_HTYPE_STMT);
     fprintf (stderr,"Customer loaded for horizontal partitioning\n");
   }
   else
   {
     fprintf (stderr,"Customer not loaded for horizontal partitioning\n");
   }
   begin_time = gettime ();
   begin_cpu = getcpu ();

   cid = 0;
   cdid = 1;
   cwid = bware;
   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;
           cwid++;                 /* shift ware cycle */
         }
       }
       c_id[i] = cid;
       c_d_id[i] = cdid;
```

```
        c_w_id[i] = cwid;
        if (cid <= 1000)
          randlastname(c_last[i], cid - 1);
        else
          randlastname(c_last[i], NURand(255, 0, 999, CNUM1));
        c_credit[i][1] = 'C';
        if (lrand48() % 10)
          c_credit[i][0] = 'G';
        else
          c_credit[i][0] = 'B';
        c_discount[i] = (float)((lrand48() % 5001) * 0.0001);
        randstr(c_first[i], 8, 16);
        randstr(c_street_1[i], 10, 20);
        randstr(c_street_2[i], 10, 20);
        randstr(c_city[i], 10, 20);
        randstr(str2, 2, 2);
        randnum(num9, 9);
        num9[4] = num9[5] = num9[6] = num9[7] = num9[8] = '1';
        randnum(num16, 16);
        randstr(c_data[i], 300, 500);

        if (gen) {
          printf("%d %d %d %s OE %s %s %s %s %s %s %s %s %cC 5000000 %6.4f -1000 1000 1 0
%s\n",
                 cid, cdid, cwid, c_first[i], c_last[i],
                 c_street_1[i], c_street_2[i], c_city[i], str2, num9,
                 num16, sdate, c_credit[i][0], c_discount[i], c_data[i]);
        }
        else {
          strncpy(c_state[i], str2, 2);
          strncpy(c_zip[i], num9, 9);
          strncpy(c_phone[i], num16, 16);
        }
      }

      if (gen) {
        fflush(stdout);
      }
      else {
        status = OCIStmtExecute(tpcsvc, curc, errhp, (ub4) 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, end_time - begin_time, end_cpu - begin_cpu);
    if (getenv("tpcc_hash_overflow")) {
      fprintf(stderr,"Hash overflow is disabled\n");
      OCIHandleAlloc(tpcenv, (dvoid **)&curi, OCI_HTYPE_STMT, 0, (dvoid**)0);
      sprintf((char *) stmbuf, SQLTXTDIHA);
      OCIStmtPrepare(curi, errhp, stmbuf, strlen((char *)stmbuf),
                     OCI_NTV_SYNTAX, OCI_DEFAULT);
      OCIERROR(errhp,OCIStmtExecute(tpcsvc, curi, errhp,1,0,0,0,OCI_DEFAULT));
      OCIHandleFree(curi, OCI_HTYPE_STMT);
    }
  }


/*---------------------------------------------------------------+
| Load the ITEM table.                                           |
+---------------------------------------------------------------*/

  if (do_A || do_i) {
    nrows = ITEMFAC;

    fprintf(stderr, "Loading/generating item: (%d rows)\n   ", nrows);

    begin_time = gettime();
    begin_cpu = getcpu();

    loopcount = 0;

    for (row = 0; row < nrows; ) {
      for (i = 0; i < ITEMARR; i++, row++) {
        i_im_id[i] = (lrand48() % 10000) + 1;
        i_price[i] = ((lrand48() % 9901) + 100);
        randstr(i_name[i], 14, 24);
        randdatastr(i_data[i], 26, 50);

        if (gen) {
          printf("%d %d %s %d %s\n", row + 1, i_im_id[i], i_name[i],
                 i_price[i], i_data[i]);
        }
        else {
          i_id[i] = row + 1;
        }
      }

      if (gen) {
        fflush(stdout);
      }
      else {
        status = OCIStmtExecute(tpcsvc, curi, errhp, (ub4) ITEMARR, (ub4) 0,
                 (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                 (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
        if (status != OCI_SUCCESS) {
                                  fprintf(stderr, "Aborted at i_id %d\n", i_id[0]);
          OCIERROR(errhp, status);
          quit();
          exit(1);
        }
      }

      if ((++loopcount) % 50)
        fprintf(stderr, ".");
      else
        fprintf(stderr, " %d rows committed\n   ", row);
    }

    end_time = gettime();
    end_cpu = getcpu();
    fprintf(stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f cpu)\n\n",
            nrows, end_time - begin_time, end_cpu - begin_cpu);
  }


/*---------------------------------------------------------------+
| Load the STOCK table.                                          |
+---------------------------------------------------------------*/

  if (do_A || do_s) {

    nrows = (eware - bware + 1) * STOCFAC;

    fprintf(stderr, "Loading/generating stock: w%d - w%d (%d rows)\n   ",
            bware, eware, nrows);

    begin_time = gettime();
    begin_cpu = getcpu();

    sid = 0;
    swid = bware;
    loopcount = 0;

    for (row = 0; row < nrows; ) {
      /* added row < nrows condition on next line - alex.ni */
                for (i = 0; (i < STOCARR) && (row < nrows); i++, row++) {
        if (++sid > STOCFAC) {        /* cheap mod */
          sid = 1;
          swid++;
        }
        s_quantity[i] = (lrand48() % 91) + 10;
        randstr(str24[0], 24, 24);
        randstr(str24[1], 24, 24);
        randstr(str24[2], 24, 24);
        randstr(str24[3], 24, 24);
        randstr(str24[4], 24, 24);
        randstr(str24[5], 24, 24);
        randstr(str24[6], 24, 24);
        randstr(str24[7], 24, 24);
        randstr(str24[8], 24, 24);
        randstr(str24[9], 24, 24);
        randdatastr(s_data[i], 26, 50);

        if (gen) {
          printf("%d %d %d %s %s %s %s %s %s %s %s %s 0 0 0  %s\n",
                 sid, swid, s_quantity[i], str24[0], str24[1], str24[2],
                 str24[3], str24[4], str24[5], str24[6], str24[7],
                 str24[8], str24[9], s_data[i]);
        }
        else {
          s_i_id[i] = sid;
          s_w_id[i] = swid;
          strncpy(s_dist_01[i], str24[0], 24);
          strncpy(s_dist_02[i], str24[1], 24);
          strncpy(s_dist_03[i], str24[2], 24);
          strncpy(s_dist_04[i], str24[3], 24);
          strncpy(s_dist_05[i], str24[4], 24);
          strncpy(s_dist_06[i], str24[5], 24);
          strncpy(s_dist_07[i], str24[6], 24);
          strncpy(s_dist_08[i], str24[7], 24);
          strncpy(s_dist_09[i], str24[8], 24);
          strncpy(s_dist_10[i], str24[9], 24);
        }
      }

      if (gen) {
        fflush(stdout);
      }
      else {
                  /* Changed to STOCKARR to i - alex.ni */
        status = OCIStmtExecute(tpcsvc, curs, errhp, (ub4) i, (ub4) 0,
```

```
                (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
      if (status != OCI_SUCCESS) {
                    fprintf (stderr, "Aborted at w_id %d, s_i_id %d\n", s_w_id[0], s_i_id[0]);
          OCIERROR(errhp, status);
          quit ();
          exit (1);
      }
    }

    if ((++loopcount) % 50)
      fprintf (stderr, ".");
    else
      fprintf (stderr, " %d rows committed\n   ", row);
  }

  end_time = gettime ();
  end_cpu = getcpu ();
  fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f cpu)\n\n",
        nrows, end_time - begin_time, end_cpu - begin_cpu);
}

/*-------------------------------------------------------------+
| Load the STOCK table (cluster around s_i_id).                |
+-------------------------------------------------------------*/

 if (do_S) {

  nrows = (eitem - bitem + 1) * (eware - bware + 1);

  fprintf (stderr, "Loading/generating stock: i%d - i%d, w%d - w%d (%d rows)\n   ",
        bitem, eitem, bware, eware, nrows);

  begin_time = gettime ();
  begin_cpu = getcpu ();

  sid = bitem;
  swid = bware - 1;
  loopcount = 0;

  for (row = 0; row < nrows; ) {
    for (i = 0; i < STOCARR; i++, row++) {
      if (++swid > eware) {          /* cheap mod */
        swid = bware;
        sid++;
      }
      s_quantity[i] = (lrand48 () % 91) + 10;
      randstr (str24[0], 24, 24);
      randstr (str24[1], 24, 24);
      randstr (str24[2], 24, 24);
      randstr (str24[3], 24, 24);
      randstr (str24[4], 24, 24);
      randstr (str24[5], 24, 24);
      randstr (str24[6], 24, 24);
      randstr (str24[7], 24, 24);
      randstr (str24[8], 24, 24);
      randstr (str24[9], 24, 24);
      randdatastr (s_data[i], 26, 50);

      if (gen) {
        printf ("'%d %d %d %s %s %s %s %s %s %s %s %s 0 0 0  %s\n",
              sid, swid, s_quantity[i], str24[0], str24[1], str24[2],
              str24[3], str24[4], str24[5], str24[6], str24[7],
              str24[8], str24[9], s_data[i]);
      }
      else {
        s_i_id[i] = sid;
        s_w_id[i] = swid;
        strncpy (s_dist_01[i], str24[0], 24);
        strncpy (s_dist_02[i], str24[1], 24);
        strncpy (s_dist_03[i], str24[2], 24);
        strncpy (s_dist_04[i], str24[3], 24);
        strncpy (s_dist_05[i], str24[4], 24);
        strncpy (s_dist_06[i], str24[5], 24);
        strncpy (s_dist_07[i], str24[6], 24);
        strncpy (s_dist_08[i], str24[7], 24);
        strncpy (s_dist_09[i], str24[8], 24);
        strncpy (s_dist_10[i], str24[9], 24);
      }
    }

    if (gen) {
      fflush (stdout);
    }
    else {
      status =  OCIStmtExecute(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, end_time - begin_time, end_cpu - begin_cpu);
 }

/*-------------------------------------------------------------+
| Load the HISTORY table.                                      |
+-------------------------------------------------------------*/

 if (do_A || do_h) {
  nrows = (eware - bware + 1) * HISTFAC;

  fprintf (stderr, "Loading/generating history: w%d - w%d (%d rows)\n   ",
        bware, eware, nrows);

  begin_time = gettime ();
  begin_cpu = getcpu ();

  cid = 0;
  cdid = 1;
  cwid = bware;
  loopcount = 0;

  for (row = 0; row < nrows; ) {
    for (i = 0; i < HISTARR; i++, row++) {
      cid++;
      if (cid > CUSTFAC) {        /* cycle cust id */
        cid = 1;                  /* cheap mod */
        cdid++;                   /* shift district cycle */
        if (cdid > DISTFAC) {
          cdid = 1;
          cwid++;                 /* shift warehouse cycle */
        }
      }
      h_c_id[i] = cid;
      h_d_id[i] = cdid;
      h_w_id[i] = cwid;
      randstr (h_data[i], 12, 24);
      if (gen) {
        printf ("'%d %d %d %d %s 1000  %s\n", cid, cdid, cwid, cdid,
              cwid, sdate, h_data[i]);
      }
    }

    if (gen) {
      fflush (stdout);
    }
    else {
      status =  OCIStmtExecute(tpcsvc, curh, errhp, (ub4) HISTARR, (ub4) 0,
                (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
      if (status != OCI_SUCCESS) {
                                  fprintf (stderr, "Aborted at w_id %d, d_id %d, c_id %d\n",
                h_w_id[0], h_d_id[0], h_c_id[0]);
          OCIERROR(errhp, status);
          quit ();
          exit (1);
      }
    }

    if ((++loopcount) % 50)
      fprintf (stderr, ".");
    else
      fprintf (stderr, " %d rows committed\n   ", row);
  }

  end_time = gettime ();
  end_cpu = getcpu ();
  fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f cpu)\n\n",
        nrows, end_time - begin_time, end_cpu - begin_cpu);
 }

/*-------------------------------------------------------------+
| Load the ORDERS and ORDER-LINE table.                        |
+-------------------------------------------------------------*/

 if (do_A || do_o) {

  int batch_olcnt;

  nrows = (eware - bware + 1) * ORDEFAC * DISTFAC;

  fprintf (stderr, "Loading/generating orders and order-line: w%d - w%d (%d ord, ~%d ordl)\n   ",
        bware, eware, nrows, nrows * 10);

  begin_time = gettime ();
  begin_cpu = getcpu ();

  cid = 0;
  cdid = 1;
  cwid = bware;
  loopcount = 0;

  for (row = 0; row < nrows; ) {
```

```
  batch_olcnt = 0;

  for (i = 0; i < ORDEARR; i++, row++) {
    cid++;
    if (cid > ORDEFAC) {        /* cycle cust id */
      cid = 1;                  /* cheap mod */
      cdid++;                   /* shift district cycle*/
      if (cdid > DISTFAC) {
        cdid = 1;
        cwid++;                 /* shift warehouse cycle*/
      }
    }
    o_carrier_id[i] = lrand48 () % 10 + 1;
    o_ol_cnt[i] = olcnt = lrand48 () % 11 + 5;

    if (gen) {
      if (cid < 2101) {
        printf ("%d %d %d %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 %s 11 %d 1\n", cid, cdid, cwid,
              randperm3000[cid - 1], sdate, o_ol_cnt[i]);
      }
    }
    else {
      o_id[i] = cid;
      o_d_id[i] = cdid;
      o_w_id[i] = cwid;
      o_c_id[i] = randperm3000[cid - 1];
      if (cid >= 2101 ) {
        o_carrier_id[i] = 11;
      }
    }

    for (j = 0; j < o_ol_cnt[i]; j++, batch_olcnt++ ) {
      ol_i_id[batch_olcnt] = sid = lrand48 () % 100000 + 1;
      if (cid < 2101)
        ol_amount[batch_olcnt] = 0;
      else
        ol_amount[batch_olcnt] = (lrand48 () % 999999 + 1) ;
      randstr (str24[j], 24, 24);

      if (gen) {
        if (cid < 2101) {
          fprintf (olfp, "%d %d %d %d %s %d %d 5 %ld %s\n", cid,
                cdid,  cwid, j + 1, sdate, ol_i_id[batch_olcnt], cwid,
                ol_amount[batch_olcnt], str24[j]);
        }
        else {
          /* Insert a default date instead of null date */
          fprintf (olfp, "%d %d %d %d 01-Jan-1811 %d %d 5 %ld %s\n", cid,
                cdid,  cwid, j + 1, ol_i_id[batch_olcnt], cwid,
                ol_amount[batch_olcnt], str24[j]);
        }
      }
      else {
        ol_o_id[batch_olcnt] = cid;
        ol_d_id[batch_olcnt] = cdid;
        ol_w_id[batch_olcnt] = cwid;
        ol_number[batch_olcnt] = j + 1;
        ol_supply_w_id[batch_olcnt]= cwid;
        strncpy (ol_dist_info[batch_olcnt], str24[j], 24);
      }
    }
    if (gen) {
      fflush (olfp);
    }
  }

  o_cnt =  ORDEARR;
  ol_cnt =  batch_olcnt;

  for (j = 0; j < batch_olcnt; j++) {
    ol_o_id_len[j] = sizeof(int);
    ol_d_id_len[j] = sizeof(int);
    ol_w_id_len[j] = sizeof(int);
    ol_number_len[j] = sizeof(int);
    ol_i_id_len[j]= sizeof(int);
    ol_supply_w_id_len[j]= sizeof(int);
    ol_dist_info_len[j]= 24;
    ol_amount_len[j] = sizeof(int);
  }
  for (j = batch_olcnt; j < 15*ORDEARR; j++) {
    ol_o_id_len[j] = 0;
    ol_d_id_len[j] = 0;
    ol_w_id_len[j] = 0;
    ol_number_len[j]= 0;
    ol_i_id_len[j]= 0;
    ol_supply_w_id_len[j]= 0;
    ol_dist_info_len[j]= 0;
    ol_amount_len[j] = 0;
  }

  o_id_clen = ORDEARR;
  o_d_id_clen = ORDEARR;
  o_w_id_clen = ORDEARR;
  o_c_id_clen = ORDEARR;
  o_carrier_id_clen = ORDEARR;
  o_ol_cnt_clen = ORDEARR;

  ol_o_id_clen = batch_olcnt;
  ol_d_id_clen = batch_olcnt;
  ol_w_id_clen = batch_olcnt;
  ol_number_clen = batch_olcnt;
  ol_i_id_clen = batch_olcnt;
  ol_supply_w_id_clen= batch_olcnt;
  ol_dist_info_clen = batch_olcnt;
  ol_amount_clen = batch_olcnt;

  OCIERROR(errhp, OCIStmtExecute(tpcsvc, curo1, errhp, (ub4) 1, (ub4) 0,
                (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS ));

  if ((++loopcount) % 50) {
    fprintf (stderr, ".");
  } else {
    fprintf (stderr, " %d orders committed\n   ", row);
  }
 }

 end_time = gettime ();
 end_cpu = getcpu ();
 fprintf (stderr, "Done.  %d orders loaded/generated in %10.2f sec. (%10.2f cpu)\n\n",
       nrows, end_time - begin_time, end_cpu - begin_cpu);
}

/*---------------------------------------------------------------+
| Load the NEW-ORDER table.                                      |
+---------------------------------------------------------------*/

if (do_A || do_n) {
 nrows = (eware - bware + 1) * NEWOFAC * DISTFAC;

 fprintf (stderr, "Loading/generating new-order: w%d - w%d (%d rows)\n   ",
       bware, eware, nrows);

 begin_time = gettime ();
 begin_cpu = getcpu ();

 cid = 0;
 cdid = 1;
 cwid = bware;
 loopcount = 0;

 for (row = 0; row <  nrows; ) {
   for (i = 0; i < NEWOARR; i++, row++) {
     cid++;
     if (cid > NEWOFAC) {
       cid = 1;
       cdid++;
       if (cdid > DISTFAC) {
         cdid = 1;
         cwid++;
       }
     }

     if (gen) {
       printf ("%d %d %d\n", cid + 2100, cdid, cwid);
     }
     else {
       no_o_id[i] = cid + 2100;
       no_d_id[i] = cdid;
       no_w_id[i] = cwid;
     }
   }

   if (gen) {
     fflush (stdout);
   }
   else {
     status =  OCIStmtExecute(tpcsvc, curno, errhp, (ub4) NEWOARR, (ub4) 0,
                   (CONST OCISnapshot*) 0, (OCISnapshot*) 0,
                   (ub4) OCI_DEFAULT | OCI_COMMIT_ON_SUCCESS);
     if (status != OCI_SUCCESS) {
                   fprintf (stderr, "Aborted at w_id %d, d_id %d, o_id %d\n", cwid, cdid, cid +
2100);
       OCIERROR(errhp, status);
       quit ();
       exit (1);
     }
   }

   if ((++loopcount) % 45)
     fprintf (stderr, ".");
   else
     fprintf (stderr, " %d rows committed\n   ", row);
 }

 end_time = gettime ();
 end_cpu = getcpu ();
 fprintf (stderr, "Done.  %d rows loaded/generated in %10.2f sec. (%10.2f cpu)\n\n",
       nrows, end_time - begin_time, end_cpu - begin_cpu);
}

/*---------------------------------------------------------------+
```

```
| clean up and exit.                                                    |
+---------------------------------------------------------------------*/

  if (olfp)
    fclose (olfp);
  if (!gen)
    quit ();
  exit (0);

}

void initperm()
{
  int i;
  int pos;
  int temp;

  /* init randperm3000 */

  for (i = 0; i < 3000; i++)
    randperm3000[i] = i + 1;
  for (i = 3000; i > 0; i--) {
    pos = lrand48 () % i;
    temp = randperm3000[i - 1];
    randperm3000[i - 1] = randperm3000[pos];
    randperm3000[pos] = temp;
  }
}

void randstr (str, x, y)
char *str;
int x;
int y;
{
  int i, j;
  int len;

  len = (lrand48 () % (y - x + 1)) + x;
  for (i = 0; i < len; i++) {
    j = lrand48 () % 62;
    if (j < 26)
      str[i] = (char) (j + 'a');
    else if (j < 52)
      str[i] = (char) (j - 26 + 'A');
    else
      str[i] = (char) (j - 52 + '0');
  }
  str[len] = '\0';

}

void randdatastr (str, x, y)
char *str;
int x;
int y;
{
  int i, j;
  int len;
  int pos;

  len = (lrand48 () % (y - x + 1)) + x;
  for (i = 0; i < len; i++) {
    j = lrand48 () % 62;
    if (j < 26)
      str[i] = (char) (j + 'a');
    else if (j < 52)
      str[i] = (char) (j - 26 + 'A');
    else
      str[i] = (char) (j - 52 + '0');
  }
  str[len] = '\0';
  if ((lrand48 () % 10) == 0) {
    pos = (lrand48 () % (len - 8));
    str[pos] = 'O';
    str[pos + 1] = 'R';
    str[pos + 2] = 'I';
    str[pos + 3] = 'G';
    str[pos + 4] = 'I';
    str[pos + 5] = 'N';
    str[pos + 6] = 'A';
    str[pos + 7] = 'L';
  }
}

void randnum (str, len)
char *str;
int len;
{
  int i;

  for (i = 0; i < len; i++)
    str[i] = (char) (lrand48 () % 10 + '0');
  str[len] = '\0';

}

void randlastname (str, id)
char *str;
int id;
{
  id = id % 1000;
  strcpy (str, lastname[id / 100]);
  strcat (str, lastname[(id / 10) % 10]);
  strcat (str, lastname[id % 10]);
}

int NURand (A, x, y, cnum)
int A, x, y, cnum;
{
  int a, b;

  a = lrand48 () % (A + 1);
  b = (lrand48 () % (y - x + 1)) + x;
  return ((((a | b) + cnum) % (y - x + 1)) + x);

}

void sysdate (sdate)
char *sdate;
{
  time_t tp;
  struct tm *tmptr;

  time (&tp);
  tmptr = localtime (&tp);
  strftime (sdate, 29, "%d-%b-%Y", tmptr);
}

int ocierror(fname, lineno, errhp, status)
char *fname;
int lineno;
OCIError *errhp;
sword status;
{
 text errbuf[512];
 sb4 errcode;
 sb4 lstat;
 ub4 recno=2;

 switch (status) {
 case OCI_SUCCESS:
   break;
 case OCI_SUCCESS_WITH_INFO:
   fprintf(stderr,"Module %s Line %d\n", fname, lineno);
   fprintf(stderr,"Error - OCI_SUCCESS_WITH_INFO\n");
   lstat = OCIErrorGet (errhp, recno++, (text *) NULL, &errcode, errbuf,
                        (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
   fprintf(stderr,"Error - %s\n", errbuf);
   break;
 case OCI_NEED_DATA:
   fprintf(stderr,"Module %s Line %d\n", fname, lineno);
   fprintf(stderr,"Error - OCI_NEED_DATA\n");
   return (IRRECERR);
 case OCI_NO_DATA:
   fprintf(stderr,"Module %s Line %d\n", fname, lineno);
   fprintf(stderr,"Error - OCI_NO_DATA\n");
   return (IRRECERR);
 case OCI_ERROR:
   lstat = OCIErrorGet (errhp, (ub4) 1,
                                (text *) NULL, &errcode, errbuf,
                                        (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
   if (errcode == NOT_SERIALIZABLE) return (errcode);
   if (errcode == SNAPSHOT_TOO_OLD) return (errcode);
   while (lstat != OCI_NO_DATA)
   {
     fprintf(stderr,"Module %s Line %d\n", fname, lineno);
     fprintf(stderr,"Error - %s\n", errbuf);
     lstat = OCIErrorGet (errhp, recno++, (text *) NULL, &errcode, errbuf,
                          (ub4) sizeof(errbuf), OCI_HTYPE_ERROR);
   }
   return (errcode);
 case OCI_INVALID_HANDLE:
   fprintf(stderr,"Module %s Line %d\n", fname, lineno);
   fprintf(stderr,"Error - OCI_INVALID_HANDLE\n");
   exit(-1);
 case OCI_STILL_EXECUTING:
   fprintf(stderr,"Module %s Line %d\n", fname, lineno);
   fprintf(stderr,"Error - OCI_STILL_EXECUTE\n");
   return (IRRECERR);
 case OCI_CONTINUE:
   fprintf(stderr,"Module %s Line %d\n", fname, lineno);
   fprintf(stderr,"Error - OCI_CONTINUE\n");
   return (IRRECERR);
 default:
   fprintf(stderr,"Module %s Line %d\n", fname, lineno);
   fprintf(stderr,"Status - %s\n", status);
   return (IRRECERR);
 }
 return (RECOVERR);
}
```

Appendix D: RTE Scripts

D.1 RTE Parameters

```
/* For Oracle in the tpccload program C_LAST =1.              */
/* C-Delta be the difference between C-LOAD and C-Run.        */
/* C-Delta must be a value between 65..119 including the      */
/* values of 65 and 119 and excluding the value of 96 and 112*/

#define MASTER_NUM1 0
#define MASTER_NUM2 0
#define MASTER_NUM3 0
#define MASTER_NUM4 0
#define MASTER_NUM5 1
#define MASTER_NUM6 0
#define MASTER_NUM7 0
#define MASTER_NUM8 0
#define MASTER_NUM9 0
#define MASTER_NUM10 0

#if MASTER_NUM1
MASTER "tpccm1"
#elif MASTER_NUM2
MASTER "tpccm2"
#elif MASTER_NUM3
MASTER "tpccm3"
#elif MASTER_NUM4
MASTER "tpccm4"
#elif MASTER_NUM5
MASTER "tpccm5"
#elif MASTER_NUM6
MASTER "tpccm6"
#elif MASTER_NUM7
MASTER "tpccm7"
#elif MASTER_NUM8
MASTER "tpccm8"
#elif MASTER_NUM9
MASTER "tpccm9"
#elif MASTER_NUM10
MASTER "tpccm10"
#endif
/*---- SUT ------------------------------------------------*/
SUT="sut"
/*---------------------------------------------------------*/
LASTC=86
MEASUREMENT="1"
WAREHOUSES=60768
/*----- SLAVES ---------------------------------------------*/

#if MASTER_NUM1
SLAVES driver1a, driver1b, driver1c, driver1d, driver2a, driver2b, driver2c, driver2d, driver3a, driver3b,
driver3c, driver3d, driver4a, driver4b, driver4c, driver4d, driver5a, driver5b, driver5c, driver5d, driver6a,
driver6b, driver6c, driver6d
#elif MASTER_NUM2
SLAVES driver7a, driver7b, driver7c, driver7d, driver8a, driver8b, driver8c, driver8d, driver9a, driver9b,
driver9c, driver9d, driver10a, driver10b, driver10c, driver10d, driver11a, driver11b, driver11c, driver11d,
driver12a, driver12b, driver12c, driver12d
#elif MASTER_NUM3
SLAVES driver13a, driver13b, driver13c, driver13d, driver14a, driver14b, driver14c, driver14d, driver15a,
driver15b, driver15c, driver15d, driver16a, driver16b, driver16c, driver16d, driver17a, driver17b, driver17c,
driver17d, driver18a, driver18b, driver18c, driver18d
#elif MASTER_NUM4
SLAVES driver19a, driver19b, driver19c, driver19d, driver20a, driver20b, driver20c, driver20d, driver21a,
driver21b, driver21c, driver21d, driver22a, driver22b, driver22c, driver22d, driver23a, driver23b, driver23c,
driver23d, driver24a, driver24b, driver24c, driver24d
#elif MASTER_NUM5
SLAVES driver25a, driver25b, driver25c, driver25d, driver26a, driver26b, driver26c, driver26d, driver27a,
driver27b, driver27c, driver27d, driver28a, driver28b, driver28c, driver28d, driver29a, driver29b, driver29c,
driver29d, driver30a, driver30b, driver30c, driver30d
#elif MASTER_NUM6
SLAVES driver31a, driver31b, driver31c, driver31d, driver32a, driver32b, driver32c, driver32d, driver33a,
driver33b, driver33c, driver33d, driver34a, driver34b, driver34c, driver34d, driver35a, driver35b, driver35c,
driver35d, driver36a, driver36b, driver36c, driver36d
#elif MASTER_NUM7
SLAVES driver37a, driver37b, driver37c, driver37d, driver38a, driver38b, driver38c, driver38d, driver39a,
driver39b, driver39c, driver39d, driver40a, driver40b, driver40c, driver40d, driver41a, driver41b, driver41c,
driver41d, driver42a, driver42b, driver42c, driver42d
#elif MASTER_NUM8
SLAVES driver43a, driver43b, driver43c, driver43d, driver44a, driver44b, driver44c, driver44d, driver45a,
driver45b, driver45c, driver45d, driver46a, driver46b, driver46c, driver46d, driver47a, driver47b, driver47c,
driver47d, driver48a, driver48b, driver48c, driver48d
#elif MASTER_NUM9
SLAVES driver49a, driver49b, driver49c, driver49d, driver50a, driver50b, driver50c, driver50d, driver51a,
driver51b, driver51c, driver51d, driver52a, driver52b, driver52c, driver52d, driver53a, driver53b, driver53c,
driver53d, driver54a, driver54b, driver54c, driver54d
#elif MASTER_NUM10
SLAVES driver55a, driver55b, driver55c, driver55d, driver56a, driver56b, driver56c, driver56d, driver57a,
driver57b, driver57c, driver57d, driver58a, driver58b, driver58c, driver58d, driver59a, driver59b, driver59c,
driver59d, driver60a, driver60b, driver60c, driver60d
#endif

/* ---- CLIENTS ------------------------------------------------*/
#if MASTER_NUM1
MAIN_CLIENT = client1
CLIENT_REAL = "client1 client2 client3"
#elif MASTER_NUM2
MAIN_CLIENT = client4
CLIENT_REAL = "client4 client5 client6"
#elif MASTER_NUM3
MAIN_CLIENT = client7
CLIENT_REAL = "client7 client8 client9"
#elif MASTER_NUM4
MAIN_CLIENT = client10
CLIENT_REAL = "client10 client11 client12"
#elif MASTER_NUM5
MAIN_CLIENT = client13
CLIENT_REAL = "client13 client14 client15"
#elif MASTER_NUM6
MAIN_CLIENT = client16
CLIENT_REAL = "client16 client17 client18"
#elif MASTER_NUM7
MAIN_CLIENT = client19
CLIENT_REAL = "client19 client20 client21"
#elif MASTER_NUM8
MAIN_CLIENT = client22
CLIENT_REAL = "client22 client23 client24"
#elif MASTER_NUM9
MAIN_CLIENT = client25
CLIENT_REAL = "client25 client26 client27"
#elif MASTER_NUM10
MAIN_CLIENT = client28
CLIENT_REAL = "client28 client29 client30"
#endif

/*---- more client ctuff ------------------------------------*/
#if MASTER_NUM1
CLIENT client11a oracle orif1db
CLIENT client12a oracle orif1db
CLIENT client11b oracle orif1db
CLIENT client12b oracle orif1db
CLIENT client11c oracle orif1db
CLIENT client12c oracle orif1db
CLIENT client11d oracle orif1db
CLIENT client12d oracle orif1db
CLIENT client11e oracle orif1db
CLIENT client12e oracle orif1db
CLIENT client11f oracle orif1db
CLIENT client12f oracle orif1db
CLIENT client11g oracle orif1db
CLIENT client12g oracle orif1db
CLIENT client11h oracle orif1db
CLIENT client12h oracle orif1db
CLIENT client11i oracle orif1db
CLIENT client12i oracle orif1db
CLIENT client11j oracle orif1db
CLIENT client12j oracle orif1db
CLIENT client11k oracle orif1db
CLIENT client12k oracle orif1db
CLIENT client11l oracle orif1db
CLIENT client12l oracle orif1db

CLIENT client21a oracle orif1db
CLIENT client22a oracle orif1db
CLIENT client21b oracle orif1db
CLIENT client22b oracle orif1db
CLIENT client21c oracle orif1db
CLIENT client22c oracle orif1db
CLIENT client21d oracle orif1db
CLIENT client22d oracle orif1db
CLIENT client21e oracle orif1db
CLIENT client22e oracle orif1db
CLIENT client21f oracle orif1db
CLIENT client22f oracle orif1db
CLIENT client21g oracle orif1db
CLIENT client22g oracle orif1db
CLIENT client21h oracle orif1db
CLIENT client22h oracle orif1db
CLIENT client21i oracle orif1db
CLIENT client22i oracle orif1db
CLIENT client21j oracle orif1db
CLIENT client22j oracle orif1db
CLIENT client21k oracle orif1db
CLIENT client22k oracle orif1db
CLIENT client21l oracle orif1db
CLIENT client22l oracle orif1db

CLIENT client31a oracle orif1db
CLIENT client32a oracle orif1db
CLIENT client31b oracle orif1db
CLIENT client32b oracle orif1db
CLIENT client31c oracle orif1db
CLIENT client32c oracle orif1db
CLIENT client31d oracle orif1db
CLIENT client32d oracle orif1db
CLIENT client31e oracle orif1db
CLIENT client32e oracle orif1db
CLIENT client31f oracle orif1db
CLIENT client32f oracle orif1db
CLIENT client31g oracle orif1db
```

```
CLIENT client32g oracle orif1db
CLIENT client31h oracle orif1db
CLIENT client32h oracle orif1db
CLIENT client31i oracle orif1db
CLIENT client32i oracle orif1db
CLIENT client31j oracle orif1db
CLIENT client32j oracle orif1db
CLIENT client31k oracle orif1db
CLIENT client32k oracle orif1db
CLIENT client31l oracle orif1db
CLIENT client32l oracle orif1db

#elif MASTER_NUM2
CLIENT client41a oracle orif1db
CLIENT client42a oracle orif1db
CLIENT client41b oracle orif1db
CLIENT client42b oracle orif1db
CLIENT client41c oracle orif1db
CLIENT client42c oracle orif1db
CLIENT client41d oracle orif1db
CLIENT client42d oracle orif1db
CLIENT client41e oracle orif1db
CLIENT client42e oracle orif1db
CLIENT client41f oracle orif1db
CLIENT client42f oracle orif1db
CLIENT client41g oracle orif1db
CLIENT client42g oracle orif1db
CLIENT client41h oracle orif1db
CLIENT client42h oracle orif1db
CLIENT client41i oracle orif1db
CLIENT client42i oracle orif1db
CLIENT client41j oracle orif1db
CLIENT client42j oracle orif1db
CLIENT client41k oracle orif1db
CLIENT client42k oracle orif1db
CLIENT client41l oracle orif1db
CLIENT client42l oracle orif1db

CLIENT client51a oracle orif1db
CLIENT client52a oracle orif1db
CLIENT client51b oracle orif1db
CLIENT client52b oracle orif1db
CLIENT client51c oracle orif1db
CLIENT client52c oracle orif1db
CLIENT client51d oracle orif1db
CLIENT client52d oracle orif1db
CLIENT client51e oracle orif1db
CLIENT client52e oracle orif1db
CLIENT client51f oracle orif1db
CLIENT client52f oracle orif1db
CLIENT client51g oracle orif1db
CLIENT client52g oracle orif1db
CLIENT client51h oracle orif1db
CLIENT client52h oracle orif1db
CLIENT client51i oracle orif1db
CLIENT client52i oracle orif1db
CLIENT client51j oracle orif1db
CLIENT client52j oracle orif1db
CLIENT client51k oracle orif1db
CLIENT client52k oracle orif1db
CLIENT client51l oracle orif1db
CLIENT client52l oracle orif1db

CLIENT client61a oracle orif1db
CLIENT client62a oracle orif1db
CLIENT client61b oracle orif1db
CLIENT client62b oracle orif1db
CLIENT client61c oracle orif1db
CLIENT client62c oracle orif1db
CLIENT client61d oracle orif1db
CLIENT client62d oracle orif1db
CLIENT client61e oracle orif1db
CLIENT client62e oracle orif1db
CLIENT client61f oracle orif1db
CLIENT client62f oracle orif1db
CLIENT client61g oracle orif1db
CLIENT client62g oracle orif1db
CLIENT client61h oracle orif1db
CLIENT client62h oracle orif1db
CLIENT client61i oracle orif1db
CLIENT client62i oracle orif1db
CLIENT client61j oracle orif1db
CLIENT client62j oracle orif1db
CLIENT client61k oracle orif1db
CLIENT client62k oracle orif1db
CLIENT client61l oracle orif1db
CLIENT client62l oracle orif1db

#elif MASTER_NUM3
CLIENT client71a oracle orif1db
CLIENT client72a oracle orif1db
CLIENT client71b oracle orif1db
CLIENT client72b oracle orif1db
CLIENT client71c oracle orif1db
CLIENT client72c oracle orif1db
CLIENT client71d oracle orif1db
CLIENT client72d oracle orif1db
CLIENT client71e oracle orif1db
CLIENT client72e oracle orif1db
CLIENT client71f oracle orif1db
CLIENT client72f oracle orif1db
CLIENT client71g oracle orif1db
CLIENT client72g oracle orif1db
CLIENT client71h oracle orif1db
CLIENT client72h oracle orif1db
CLIENT client71i oracle orif1db
CLIENT client72i oracle orif1db
CLIENT client71j oracle orif1db
CLIENT client72j oracle orif1db
CLIENT client71k oracle orif1db
CLIENT client72k oracle orif1db
CLIENT client71l oracle orif1db
CLIENT client72l oracle orif1db

CLIENT client81a oracle orif1db
CLIENT client82a oracle orif1db
CLIENT client81b oracle orif1db
CLIENT client82b oracle orif1db
CLIENT client81c oracle orif1db
CLIENT client82c oracle orif1db
CLIENT client81d oracle orif1db
CLIENT client82d oracle orif1db
CLIENT client81e oracle orif1db
CLIENT client82e oracle orif1db
CLIENT client81f oracle orif1db
CLIENT client82f oracle orif1db
CLIENT client81g oracle orif1db
CLIENT client82g oracle orif1db
CLIENT client81h oracle orif1db
CLIENT client82h oracle orif1db
CLIENT client81i oracle orif1db
CLIENT client82i oracle orif1db
CLIENT client81j oracle orif1db
CLIENT client82j oracle orif1db
CLIENT client81k oracle orif1db
CLIENT client82k oracle orif1db
CLIENT client81l oracle orif1db
CLIENT client82l oracle orif1db

CLIENT client91a oracle orif1db
CLIENT client92a oracle orif1db
CLIENT client91b oracle orif1db
CLIENT client92b oracle orif1db
CLIENT client91c oracle orif1db
CLIENT client92c oracle orif1db
CLIENT client91d oracle orif1db
CLIENT client92d oracle orif1db
CLIENT client91e oracle orif1db
CLIENT client92e oracle orif1db
CLIENT client91f oracle orif1db
CLIENT client92f oracle orif1db
CLIENT client91g oracle orif1db
CLIENT client92g oracle orif1db
CLIENT client91h oracle orif1db
CLIENT client92h oracle orif1db
CLIENT client91i oracle orif1db
CLIENT client92i oracle orif1db
CLIENT client91j oracle orif1db
CLIENT client92j oracle orif1db
CLIENT client91k oracle orif1db
CLIENT client92k oracle orif1db
CLIENT client91l oracle orif1db
CLIENT client92l oracle orif1db

#elif MASTER_NUM4
CLIENT client101a oracle orif1db
CLIENT client102a oracle orif1db
CLIENT client101b oracle orif1db
CLIENT client102b oracle orif1db
CLIENT client101c oracle orif1db
CLIENT client102c oracle orif1db
CLIENT client101d oracle orif1db
CLIENT client102d oracle orif1db
CLIENT client101e oracle orif1db
CLIENT client102e oracle orif1db
CLIENT client101f oracle orif1db
CLIENT client102f oracle orif1db
CLIENT client101g oracle orif1db
CLIENT client102g oracle orif1db
CLIENT client101h oracle orif1db
CLIENT client102h oracle orif1db
CLIENT client101i oracle orif1db
CLIENT client102i oracle orif1db
CLIENT client101j oracle orif1db
CLIENT client102j oracle orif1db
CLIENT client101k oracle orif1db
CLIENT client102k oracle orif1db
CLIENT client101l oracle orif1db
CLIENT client102l oracle orif1db

CLIENT client111a oracle orif1db
CLIENT client112a oracle orif1db
CLIENT client111b oracle orif1db
CLIENT client112b oracle orif1db
CLIENT client111c oracle orif1db
CLIENT client112c oracle orif1db
CLIENT client111d oracle orif1db
CLIENT client112d oracle orif1db
```

```
CLIENT client111e oracle orif1db
CLIENT client112e oracle orif1db
CLIENT client111f oracle orif1db
CLIENT client112f oracle orif1db
CLIENT client111g oracle orif1db
CLIENT client112g oracle orif1db
CLIENT client111h oracle orif1db
CLIENT client112h oracle orif1db
CLIENT client111i oracle orif1db
CLIENT client112i oracle orif1db
CLIENT client111j oracle orif1db
CLIENT client112j oracle orif1db
CLIENT client111k oracle orif1db
CLIENT client112k oracle orif1db
CLIENT client111l oracle orif1db
CLIENT client112l oracle orif1db

CLIENT client121a oracle orif1db
CLIENT client122a oracle orif1db
CLIENT client121b oracle orif1db
CLIENT client122b oracle orif1db
CLIENT client121c oracle orif1db
CLIENT client122c oracle orif1db
CLIENT client121d oracle orif1db
CLIENT client122d oracle orif1db
CLIENT client121e oracle orif1db
CLIENT client122e oracle orif1db
CLIENT client121f oracle orif1db
CLIENT client122f oracle orif1db
CLIENT client121g oracle orif1db
CLIENT client122g oracle orif1db
CLIENT client121h oracle orif1db
CLIENT client122h oracle orif1db
CLIENT client121i oracle orif1db
CLIENT client122i oracle orif1db
CLIENT client121j oracle orif1db
CLIENT client122j oracle orif1db
CLIENT client121k oracle orif1db
CLIENT client122k oracle orif1db
CLIENT client121l oracle orif1db
CLIENT client122l oracle orif1db

#elif MASTER_NUM5
CLIENT client131a oracle orif1db
CLIENT client132a oracle orif1db
CLIENT client131b oracle orif1db
CLIENT client132b oracle orif1db
CLIENT client131c oracle orif1db
CLIENT client132c oracle orif1db
CLIENT client131d oracle orif1db
CLIENT client132d oracle orif1db
CLIENT client131e oracle orif1db
CLIENT client132e oracle orif1db
CLIENT client131f oracle orif1db
CLIENT client132f oracle orif1db
CLIENT client131g oracle orif1db
CLIENT client132g oracle orif1db
CLIENT client131h oracle orif1db
CLIENT client132h oracle orif1db
CLIENT client131i oracle orif1db
CLIENT client132i oracle orif1db
CLIENT client131j oracle orif1db
CLIENT client132j oracle orif1db
CLIENT client131k oracle orif1db
CLIENT client132k oracle orif1db
CLIENT client131l oracle orif1db
CLIENT client132l oracle orif1db

CLIENT client141a oracle orif1db
CLIENT client142a oracle orif1db
CLIENT client141b oracle orif1db
CLIENT client142b oracle orif1db
CLIENT client141c oracle orif1db
CLIENT client142c oracle orif1db
CLIENT client141d oracle orif1db
CLIENT client142d oracle orif1db
CLIENT client141e oracle orif1db
CLIENT client142e oracle orif1db
CLIENT client141f oracle orif1db
CLIENT client142f oracle orif1db
CLIENT client141g oracle orif1db
CLIENT client142g oracle orif1db
CLIENT client141h oracle orif1db
CLIENT client142h oracle orif1db
CLIENT client141i oracle orif1db
CLIENT client142i oracle orif1db
CLIENT client141j oracle orif1db
CLIENT client142j oracle orif1db
CLIENT client141k oracle orif1db
CLIENT client142k oracle orif1db
CLIENT client141l oracle orif1db
CLIENT client142l oracle orif1db

CLIENT client151a oracle orif1db
CLIENT client152a oracle orif1db
CLIENT client151b oracle orif1db
CLIENT client152b oracle orif1db
CLIENT client151c oracle orif1db
CLIENT client152c oracle orif1db
CLIENT client151d oracle orif1db
CLIENT client152d oracle orif1db
CLIENT client151e oracle orif1db
CLIENT client152e oracle orif1db
CLIENT client151f oracle orif1db
CLIENT client152f oracle orif1db
CLIENT client151g oracle orif1db
CLIENT client152g oracle orif1db
CLIENT client151h oracle orif1db
CLIENT client152h oracle orif1db
CLIENT client151i oracle orif1db
CLIENT client152i oracle orif1db
CLIENT client151j oracle orif1db
CLIENT client152j oracle orif1db
CLIENT client151k oracle orif1db
CLIENT client152k oracle orif1db
CLIENT client151l oracle orif1db
CLIENT client152l oracle orif1db

#elif MASTER_NUM6
CLIENT client161a oracle orif1db
CLIENT client162a oracle orif1db
CLIENT client161b oracle orif1db
CLIENT client162b oracle orif1db
CLIENT client161c oracle orif1db
CLIENT client162c oracle orif1db
CLIENT client161d oracle orif1db
CLIENT client162d oracle orif1db
CLIENT client161e oracle orif1db
CLIENT client162e oracle orif1db
CLIENT client161f oracle orif1db
CLIENT client162f oracle orif1db
CLIENT client161g oracle orif1db
CLIENT client162g oracle orif1db
CLIENT client161h oracle orif1db
CLIENT client162h oracle orif1db
CLIENT client161i oracle orif1db
CLIENT client162i oracle orif1db
CLIENT client161j oracle orif1db
CLIENT client162j oracle orif1db
CLIENT client161k oracle orif1db
CLIENT client162k oracle orif1db
CLIENT client161l oracle orif1db
CLIENT client162l oracle orif1db

CLIENT client171a oracle orif1db
CLIENT client172a oracle orif1db
CLIENT client171b oracle orif1db
CLIENT client172b oracle orif1db
CLIENT client171c oracle orif1db
CLIENT client172c oracle orif1db
CLIENT client171d oracle orif1db
CLIENT client172d oracle orif1db
CLIENT client171e oracle orif1db
CLIENT client172e oracle orif1db
CLIENT client171f oracle orif1db
CLIENT client172f oracle orif1db
CLIENT client171g oracle orif1db
CLIENT client172g oracle orif1db
CLIENT client171h oracle orif1db
CLIENT client172h oracle orif1db
CLIENT client171i oracle orif1db
CLIENT client172i oracle orif1db
CLIENT client171j oracle orif1db
CLIENT client172j oracle orif1db
CLIENT client171k oracle orif1db
CLIENT client172k oracle orif1db
CLIENT client171l oracle orif1db
CLIENT client172l oracle orif1db

CLIENT client181a oracle orif1db
CLIENT client182a oracle orif1db
CLIENT client181b oracle orif1db
CLIENT client182b oracle orif1db
CLIENT client181c oracle orif1db
CLIENT client182c oracle orif1db
CLIENT client181d oracle orif1db
CLIENT client182d oracle orif1db
CLIENT client181e oracle orif1db
CLIENT client182e oracle orif1db
CLIENT client181f oracle orif1db
CLIENT client182f oracle orif1db
CLIENT client181g oracle orif1db
CLIENT client182g oracle orif1db
CLIENT client181h oracle orif1db
CLIENT client182h oracle orif1db
CLIENT client181i oracle orif1db
CLIENT client182i oracle orif1db
CLIENT client181j oracle orif1db
CLIENT client182j oracle orif1db
CLIENT client181k oracle orif1db
CLIENT client182k oracle orif1db
CLIENT client181l oracle orif1db
CLIENT client182l oracle orif1db

#elif MASTER_NUM7
CLIENT client191a oracle orif1db
CLIENT client192a oracle orif1db
CLIENT client191b oracle orif1db
```

CLIENT client192b oracle orif1db
CLIENT client191c oracle orif1db
CLIENT client192c oracle orif1db
CLIENT client191d oracle orif1db
CLIENT client192d oracle orif1db
CLIENT client191e oracle orif1db
CLIENT client192e oracle orif1db
CLIENT client191f oracle orif1db
CLIENT client192f oracle orif1db
CLIENT client191g oracle orif1db
CLIENT client192g oracle orif1db
CLIENT client191h oracle orif1db
CLIENT client192h oracle orif1db
CLIENT client191i oracle orif1db
CLIENT client192i oracle orif1db
CLIENT client191j oracle orif1db
CLIENT client192j oracle orif1db
CLIENT client191k oracle orif1db
CLIENT client192k oracle orif1db
CLIENT client191l oracle orif1db
CLIENT client192l oracle orif1db

CLIENT client201a oracle orif1db
CLIENT client202a oracle orif1db
CLIENT client201b oracle orif1db
CLIENT client202b oracle orif1db
CLIENT client201c oracle orif1db
CLIENT client202c oracle orif1db
CLIENT client201d oracle orif1db
CLIENT client202d oracle orif1db
CLIENT client201e oracle orif1db
CLIENT client202e oracle orif1db
CLIENT client201f oracle orif1db
CLIENT client202f oracle orif1db
CLIENT client201g oracle orif1db
CLIENT client202g oracle orif1db
CLIENT client201h oracle orif1db
CLIENT client202h oracle orif1db
CLIENT client201i oracle orif1db
CLIENT client202i oracle orif1db
CLIENT client201j oracle orif1db
CLIENT client202j oracle orif1db
CLIENT client201k oracle orif1db
CLIENT client202k oracle orif1db
CLIENT client201l oracle orif1db
CLIENT client202l oracle orif1db

CLIENT client211a oracle orif1db
CLIENT client212a oracle orif1db
CLIENT client211b oracle orif1db
CLIENT client212b oracle orif1db
CLIENT client211c oracle orif1db
CLIENT client212c oracle orif1db
CLIENT client211d oracle orif1db
CLIENT client212d oracle orif1db
CLIENT client211e oracle orif1db
CLIENT client212e oracle orif1db
CLIENT client211f oracle orif1db
CLIENT client212f oracle orif1db
CLIENT client211g oracle orif1db
CLIENT client212g oracle orif1db
CLIENT client211h oracle orif1db
CLIENT client212h oracle orif1db
CLIENT client211i oracle orif1db
CLIENT client212i oracle orif1db
CLIENT client211j oracle orif1db
CLIENT client212j oracle orif1db
CLIENT client211k oracle orif1db
CLIENT client212k oracle orif1db
CLIENT client211l oracle orif1db
CLIENT client212l oracle orif1db

#elif MASTER_NUM8
CLIENT client221a oracle orif1db
CLIENT client222a oracle orif1db
CLIENT client221b oracle orif1db
CLIENT client222b oracle orif1db
CLIENT client221c oracle orif1db
CLIENT client222c oracle orif1db
CLIENT client221d oracle orif1db
CLIENT client222d oracle orif1db
CLIENT client221e oracle orif1db
CLIENT client222e oracle orif1db
CLIENT client221f oracle orif1db
CLIENT client222f oracle orif1db
CLIENT client221g oracle orif1db
CLIENT client222g oracle orif1db
CLIENT client221h oracle orif1db
CLIENT client222h oracle orif1db
CLIENT client221i oracle orif1db
CLIENT client222i oracle orif1db
CLIENT client221j oracle orif1db
CLIENT client222j oracle orif1db
CLIENT client221k oracle orif1db
CLIENT client222k oracle orif1db
CLIENT client221l oracle orif1db
CLIENT client222l oracle orif1db

CLIENT client231a oracle orif1db
CLIENT client232a oracle orif1db
CLIENT client231b oracle orif1db
CLIENT client232b oracle orif1db
CLIENT client231c oracle orif1db
CLIENT client232c oracle orif1db
CLIENT client231d oracle orif1db
CLIENT client232d oracle orif1db
CLIENT client231e oracle orif1db
CLIENT client232e oracle orif1db
CLIENT client231f oracle orif1db
CLIENT client232f oracle orif1db
CLIENT client231g oracle orif1db
CLIENT client232g oracle orif1db
CLIENT client231h oracle orif1db
CLIENT client232h oracle orif1db
CLIENT client231i oracle orif1db
CLIENT client232i oracle orif1db
CLIENT client231j oracle orif1db
CLIENT client232j oracle orif1db
CLIENT client231k oracle orif1db
CLIENT client232k oracle orif1db
CLIENT client231l oracle orif1db
CLIENT client232l oracle orif1db

CLIENT client241a oracle orif1db
CLIENT client242a oracle orif1db
CLIENT client241b oracle orif1db
CLIENT client242b oracle orif1db
CLIENT client241c oracle orif1db
CLIENT client242c oracle orif1db
CLIENT client241d oracle orif1db
CLIENT client242d oracle orif1db
CLIENT client241e oracle orif1db
CLIENT client242e oracle orif1db
CLIENT client241f oracle orif1db
CLIENT client242f oracle orif1db
CLIENT client241g oracle orif1db
CLIENT client242g oracle orif1db
CLIENT client241h oracle orif1db
CLIENT client242h oracle orif1db
CLIENT client241i oracle orif1db
CLIENT client242i oracle orif1db
CLIENT client241j oracle orif1db
CLIENT client242j oracle orif1db
CLIENT client241k oracle orif1db
CLIENT client242k oracle orif1db
CLIENT client241l oracle orif1db
CLIENT client242l oracle orif1db

#elif MASTER_NUM9
CLIENT client251a oracle orif1db
CLIENT client252a oracle orif1db
CLIENT client251b oracle orif1db
CLIENT client252b oracle orif1db
CLIENT client251c oracle orif1db
CLIENT client252c oracle orif1db
CLIENT client251d oracle orif1db
CLIENT client252d oracle orif1db
CLIENT client251e oracle orif1db
CLIENT client252e oracle orif1db
CLIENT client251f oracle orif1db
CLIENT client252f oracle orif1db
CLIENT client251g oracle orif1db
CLIENT client252g oracle orif1db
CLIENT client251h oracle orif1db
CLIENT client252h oracle orif1db
CLIENT client251i oracle orif1db
CLIENT client252i oracle orif1db
CLIENT client251j oracle orif1db
CLIENT client252j oracle orif1db
CLIENT client251k oracle orif1db
CLIENT client252k oracle orif1db
CLIENT client251l oracle orif1db
CLIENT client252l oracle orif1db

CLIENT client261a oracle orif1db
CLIENT client262a oracle orif1db
CLIENT client261b oracle orif1db
CLIENT client262b oracle orif1db
CLIENT client261c oracle orif1db
CLIENT client262c oracle orif1db
CLIENT client261d oracle orif1db
CLIENT client262d oracle orif1db
CLIENT client261e oracle orif1db
CLIENT client262e oracle orif1db
CLIENT client261f oracle orif1db
CLIENT client262f oracle orif1db
CLIENT client261g oracle orif1db
CLIENT client262g oracle orif1db
CLIENT client261h oracle orif1db
CLIENT client262h oracle orif1db
CLIENT client261i oracle orif1db
CLIENT client262i oracle orif1db
CLIENT client261j oracle orif1db
CLIENT client262j oracle orif1db
CLIENT client261k oracle orif1db
CLIENT client262k oracle orif1db
CLIENT client261l oracle orif1db
CLIENT client262l oracle orif1db

```
CLIENT client271a oracle orif1db
CLIENT client272a oracle orif1db
CLIENT client271b oracle orif1db
CLIENT client272b oracle orif1db
CLIENT client271c oracle orif1db
CLIENT client272c oracle orif1db
CLIENT client271d oracle orif1db
CLIENT client272d oracle orif1db
CLIENT client271e oracle orif1db
CLIENT client272e oracle orif1db
CLIENT client271f oracle orif1db
CLIENT client272f oracle orif1db
CLIENT client271g oracle orif1db
CLIENT client272g oracle orif1db
CLIENT client271h oracle orif1db
CLIENT client272h oracle orif1db
CLIENT client271i oracle orif1db
CLIENT client272i oracle orif1db
CLIENT client271j oracle orif1db
CLIENT client272j oracle orif1db
CLIENT client271k oracle orif1db
CLIENT client272k oracle orif1db
CLIENT client271l oracle orif1db
CLIENT client272l oracle orif1db

#elif MASTER_NUM10
CLIENT client281a oracle orif1db
CLIENT client282a oracle orif1db
CLIENT client281b oracle orif1db
CLIENT client282b oracle orif1db
CLIENT client281c oracle orif1db
CLIENT client282c oracle orif1db
CLIENT client281d oracle orif1db
CLIENT client282d oracle orif1db
CLIENT client281e oracle orif1db
CLIENT client282e oracle orif1db
CLIENT client281f oracle orif1db
CLIENT client282f oracle orif1db
CLIENT client281g oracle orif1db
CLIENT client282g oracle orif1db
CLIENT client281h oracle orif1db
CLIENT client282h oracle orif1db
CLIENT client281i oracle orif1db
CLIENT client282i oracle orif1db
CLIENT client281j oracle orif1db
CLIENT client282j oracle orif1db
CLIENT client281k oracle orif1db
CLIENT client282k oracle orif1db
CLIENT client281l oracle orif1db
CLIENT client282l oracle orif1db

CLIENT client291a oracle orif1db
CLIENT client292a oracle orif1db
CLIENT client291b oracle orif1db
CLIENT client292b oracle orif1db
CLIENT client291c oracle orif1db
CLIENT client292c oracle orif1db
CLIENT client291d oracle orif1db
CLIENT client292d oracle orif1db
CLIENT client291e oracle orif1db
CLIENT client292e oracle orif1db
CLIENT client291f oracle orif1db
CLIENT client292f oracle orif1db
CLIENT client291g oracle orif1db
CLIENT client292g oracle orif1db
CLIENT client291h oracle orif1db
CLIENT client292h oracle orif1db
CLIENT client291i oracle orif1db
CLIENT client292i oracle orif1db
CLIENT client291j oracle orif1db
CLIENT client292j oracle orif1db
CLIENT client291k oracle orif1db
CLIENT client292k oracle orif1db
CLIENT client291l oracle orif1db
CLIENT client292l oracle orif1db

CLIENT client301a oracle orif1db
CLIENT client302a oracle orif1db
CLIENT client301b oracle orif1db
CLIENT client302b oracle orif1db
CLIENT client301c oracle orif1db
CLIENT client302c oracle orif1db
CLIENT client301d oracle orif1db
CLIENT client302d oracle orif1db
CLIENT client301e oracle orif1db
CLIENT client302e oracle orif1db
CLIENT client301f oracle orif1db
CLIENT client302f oracle orif1db
CLIENT client301g oracle orif1db
CLIENT client302g oracle orif1db
CLIENT client301h oracle orif1db
CLIENT client302h oracle orif1db
CLIENT client301i oracle orif1db
CLIENT client302i oracle orif1db
CLIENT client301j oracle orif1db
CLIENT client302j oracle orif1db
CLIENT client301k oracle orif1db
CLIENT client302k oracle orif1db
CLIENT client301l oracle orif1db
CLIENT client302l oracle orif1db

#endif
/*----------------------------------------------------------------*/
TELNET telnet 23
SOCKET socket 199703
/* --- Sockets ------------------------------------------------*/
#if MASTER_NUM1
SOCKET_NETWORK socket1   6700 driver1a
SOCKET_NETWORK socket2   6701 driver1b
SOCKET_NETWORK socket3   6702 driver1c
SOCKET_NETWORK socket4   6703 driver1d
SOCKET_NETWORK socket5   6704 driver1a
SOCKET_NETWORK socket6   6705 driver1b
SOCKET_NETWORK socket7   6706 driver1c
SOCKET_NETWORK socket8   6707 driver1d
SOCKET_NETWORK socket9   6708 driver1a
SOCKET_NETWORK socket10  6709 driver1b
SOCKET_NETWORK socket11  6710 driver1c
SOCKET_NETWORK socket12  6711 driver1d
SOCKET_NETWORK socket13  6712 driver1a
SOCKET_NETWORK socket14  6713 driver1b
SOCKET_NETWORK socket15  6714 driver1c
SOCKET_NETWORK socket16  6715 driver1d
SOCKET_NETWORK socket17  6716 driver1a
SOCKET_NETWORK socket18  6717 driver1b
SOCKET_NETWORK socket19  6718 driver1c
SOCKET_NETWORK socket20  6719 driver1d
SOCKET_NETWORK socket21  6720 driver1a
SOCKET_NETWORK socket22  6721 driver1b
SOCKET_NETWORK socket23  6722 driver1c
SOCKET_NETWORK socket24  6723 driver1d
SOCKET_NETWORK socket25  6724 driver1a
SOCKET_NETWORK socket26  6725 driver1b
SOCKET_NETWORK socket27  6726 driver1c
SOCKET_NETWORK socket28  6727 driver1d
SOCKET_NETWORK socket29  6728 driver1a
SOCKET_NETWORK socket30  6729 driver1b
SOCKET_NETWORK socket31  6730 driver1c
SOCKET_NETWORK socket32  6731 driver1d
SOCKET_NETWORK socket33  6732 driver1a
SOCKET_NETWORK socket34  6733 driver1b
SOCKET_NETWORK socket35  6734 driver1c
SOCKET_NETWORK socket36  6735 driver1d
SOCKET_NETWORK socket37  6736 driver1a
SOCKET_NETWORK socket38  6737 driver1b
SOCKET_NETWORK socket39  6738 driver1c
SOCKET_NETWORK socket40  6739 driver1d
SOCKET_NETWORK socket41  6740 driver1a
SOCKET_NETWORK socket42  6741 driver1b
SOCKET_NETWORK socket43  6742 driver1c
SOCKET_NETWORK socket44  6743 driver1d
SOCKET_NETWORK socket45  6744 driver1a
SOCKET_NETWORK socket46  6745 driver1b
SOCKET_NETWORK socket47  6746 driver1c
SOCKET_NETWORK socket48  6747 driver1d
SOCKET_NETWORK socket49  6748 driver2a
SOCKET_NETWORK socket50  6749 driver2b
SOCKET_NETWORK socket51  6750 driver2c
SOCKET_NETWORK socket52  6751 driver2d
SOCKET_NETWORK socket53  6752 driver2a
SOCKET_NETWORK socket54  6753 driver2b
SOCKET_NETWORK socket55  6754 driver2c
SOCKET_NETWORK socket56  6755 driver2d
SOCKET_NETWORK socket57  6756 driver2a
SOCKET_NETWORK socket58  6757 driver2b
SOCKET_NETWORK socket59  6758 driver2c
SOCKET_NETWORK socket60  6759 driver2d
SOCKET_NETWORK socket61  6760 driver2a
SOCKET_NETWORK socket62  6761 driver2b
SOCKET_NETWORK socket63  6762 driver2c
SOCKET_NETWORK socket64  6763 driver2d
SOCKET_NETWORK socket65  6764 driver2a
SOCKET_NETWORK socket66  6765 driver2b
SOCKET_NETWORK socket67  6766 driver2c
SOCKET_NETWORK socket68  6767 driver2d
SOCKET_NETWORK socket69  6768 driver2a
SOCKET_NETWORK socket70  6769 driver2b
SOCKET_NETWORK socket71  6770 driver2c
SOCKET_NETWORK socket72  6771 driver2d
SOCKET_NETWORK socket73  6772 driver2a
SOCKET_NETWORK socket74  6773 driver2b
SOCKET_NETWORK socket75  6774 driver2c
SOCKET_NETWORK socket76  6775 driver2d
SOCKET_NETWORK socket77  6776 driver2a
SOCKET_NETWORK socket78  6777 driver2b
SOCKET_NETWORK socket79  6778 driver2c
SOCKET_NETWORK socket80  6779 driver2d
SOCKET_NETWORK socket81  6780 driver2a
SOCKET_NETWORK socket82  6781 driver2b
SOCKET_NETWORK socket83  6782 driver2c
SOCKET_NETWORK socket84  6783 driver2d
SOCKET_NETWORK socket85  6784 driver2a
SOCKET_NETWORK socket86  6785 driver2b
SOCKET_NETWORK socket87  6786 driver2c
SOCKET_NETWORK socket88  6787 driver2d
SOCKET_NETWORK socket89  6788 driver2a
SOCKET_NETWORK socket90  6789 driver2b
```

SOCKET_NETWORK socket91 6790 driver2c
SOCKET_NETWORK socket92 6791 driver2d
SOCKET_NETWORK socket93 6792 driver2a
SOCKET_NETWORK socket94 6793 driver2b
SOCKET_NETWORK socket95 6794 driver2c
SOCKET_NETWORK socket96 6795 driver2d
SOCKET_NETWORK socket97 6700 driver3a
SOCKET_NETWORK socket98 6701 driver3b
SOCKET_NETWORK socket99 6702 driver3c
SOCKET_NETWORK socket100 6703 driver3d
SOCKET_NETWORK socket101 6704 driver3a
SOCKET_NETWORK socket102 6705 driver3b
SOCKET_NETWORK socket103 6706 driver3c
SOCKET_NETWORK socket104 6707 driver3d
SOCKET_NETWORK socket105 6708 driver3a
SOCKET_NETWORK socket106 6709 driver3b
SOCKET_NETWORK socket107 6710 driver3c
SOCKET_NETWORK socket108 6711 driver3d
SOCKET_NETWORK socket109 6712 driver3a
SOCKET_NETWORK socket110 6713 driver3b
SOCKET_NETWORK socket111 6714 driver3c
SOCKET_NETWORK socket112 6715 driver3d
SOCKET_NETWORK socket113 6716 driver3a
SOCKET_NETWORK socket114 6717 driver3b
SOCKET_NETWORK socket115 6718 driver3c
SOCKET_NETWORK socket116 6719 driver3d
SOCKET_NETWORK socket117 6720 driver3a
SOCKET_NETWORK socket118 6721 driver3b
SOCKET_NETWORK socket119 6722 driver3c
SOCKET_NETWORK socket120 6723 driver3d
SOCKET_NETWORK socket121 6724 driver3a
SOCKET_NETWORK socket122 6725 driver3b
SOCKET_NETWORK socket123 6726 driver3c
SOCKET_NETWORK socket124 6727 driver3d
SOCKET_NETWORK socket125 6728 driver3a
SOCKET_NETWORK socket126 6729 driver3b
SOCKET_NETWORK socket127 6730 driver3c
SOCKET_NETWORK socket128 6731 driver3d
SOCKET_NETWORK socket129 6732 driver3a
SOCKET_NETWORK socket130 6733 driver3b
SOCKET_NETWORK socket131 6734 driver3c
SOCKET_NETWORK socket132 6735 driver3d
SOCKET_NETWORK socket133 6736 driver3a
SOCKET_NETWORK socket134 6737 driver3b
SOCKET_NETWORK socket135 6738 driver3c
SOCKET_NETWORK socket136 6739 driver3d
SOCKET_NETWORK socket137 6740 driver3a
SOCKET_NETWORK socket138 6741 driver3b
SOCKET_NETWORK socket139 6742 driver3c
SOCKET_NETWORK socket140 6743 driver3d
SOCKET_NETWORK socket141 6744 driver3a
SOCKET_NETWORK socket142 6745 driver3b
SOCKET_NETWORK socket143 6746 driver3c
SOCKET_NETWORK socket144 6747 driver3d
SOCKET_NETWORK socket145 6748 driver4a
SOCKET_NETWORK socket146 6749 driver4b
SOCKET_NETWORK socket147 6750 driver4c
SOCKET_NETWORK socket148 6751 driver4d
SOCKET_NETWORK socket149 6752 driver4a
SOCKET_NETWORK socket150 6753 driver4b
SOCKET_NETWORK socket151 6754 driver4c
SOCKET_NETWORK socket152 6755 driver4d
SOCKET_NETWORK socket153 6756 driver4a
SOCKET_NETWORK socket154 6757 driver4b
SOCKET_NETWORK socket155 6758 driver4c
SOCKET_NETWORK socket156 6759 driver4d
SOCKET_NETWORK socket157 6760 driver4a
SOCKET_NETWORK socket158 6761 driver4b
SOCKET_NETWORK socket159 6762 driver4c
SOCKET_NETWORK socket160 6763 driver4d
SOCKET_NETWORK socket161 6764 driver4a
SOCKET_NETWORK socket162 6765 driver4b
SOCKET_NETWORK socket163 6766 driver4c
SOCKET_NETWORK socket164 6767 driver4d
SOCKET_NETWORK socket165 6768 driver4a
SOCKET_NETWORK socket166 6769 driver4b
SOCKET_NETWORK socket167 6770 driver4c
SOCKET_NETWORK socket168 6771 driver4d
SOCKET_NETWORK socket169 6772 driver4a
SOCKET_NETWORK socket170 6773 driver4b
SOCKET_NETWORK socket171 6774 driver4c
SOCKET_NETWORK socket172 6775 driver4d
SOCKET_NETWORK socket173 6776 driver4a
SOCKET_NETWORK socket174 6777 driver4b
SOCKET_NETWORK socket175 6778 driver4c
SOCKET_NETWORK socket176 6779 driver4d
SOCKET_NETWORK socket177 6780 driver4a
SOCKET_NETWORK socket178 6781 driver4b
SOCKET_NETWORK socket179 6782 driver4c
SOCKET_NETWORK socket180 6783 driver4d
SOCKET_NETWORK socket181 6784 driver4a
SOCKET_NETWORK socket182 6785 driver4b
SOCKET_NETWORK socket183 6786 driver4c
SOCKET_NETWORK socket184 6787 driver4d
SOCKET_NETWORK socket185 6788 driver4a
SOCKET_NETWORK socket186 6789 driver4b
SOCKET_NETWORK socket187 6790 driver4c
SOCKET_NETWORK socket188 6791 driver4d
SOCKET_NETWORK socket189 6792 driver4a
SOCKET_NETWORK socket190 6793 driver4b
SOCKET_NETWORK socket191 6794 driver4c
SOCKET_NETWORK socket192 6795 driver4d
SOCKET_NETWORK socket193 6700 driver5a
SOCKET_NETWORK socket194 6701 driver5b
SOCKET_NETWORK socket195 6702 driver5c
SOCKET_NETWORK socket196 6703 driver5d
SOCKET_NETWORK socket197 6704 driver5a
SOCKET_NETWORK socket198 6705 driver5b
SOCKET_NETWORK socket199 6706 driver5c
SOCKET_NETWORK socket200 6707 driver5d
SOCKET_NETWORK socket201 6708 driver5a
SOCKET_NETWORK socket202 6709 driver5b
SOCKET_NETWORK socket203 6710 driver5c
SOCKET_NETWORK socket204 6711 driver5d
SOCKET_NETWORK socket205 6712 driver5a
SOCKET_NETWORK socket206 6713 driver5b
SOCKET_NETWORK socket207 6714 driver5c
SOCKET_NETWORK socket208 6715 driver5d
SOCKET_NETWORK socket209 6716 driver5a
SOCKET_NETWORK socket210 6717 driver5b
SOCKET_NETWORK socket211 6718 driver5c
SOCKET_NETWORK socket212 6719 driver5d
SOCKET_NETWORK socket213 6720 driver5a
SOCKET_NETWORK socket214 6721 driver5b
SOCKET_NETWORK socket215 6722 driver5c
SOCKET_NETWORK socket216 6723 driver5d
SOCKET_NETWORK socket217 6724 driver5a
SOCKET_NETWORK socket218 6725 driver5b
SOCKET_NETWORK socket219 6726 driver5c
SOCKET_NETWORK socket220 6727 driver5d
SOCKET_NETWORK socket221 6728 driver5a
SOCKET_NETWORK socket222 6729 driver5b
SOCKET_NETWORK socket223 6730 driver5c
SOCKET_NETWORK socket224 6731 driver5d
SOCKET_NETWORK socket225 6732 driver5a
SOCKET_NETWORK socket226 6733 driver5b
SOCKET_NETWORK socket227 6734 driver5c
SOCKET_NETWORK socket228 6735 driver5d
SOCKET_NETWORK socket229 6736 driver5a
SOCKET_NETWORK socket230 6737 driver5b
SOCKET_NETWORK socket231 6738 driver5c
SOCKET_NETWORK socket232 6739 driver5d
SOCKET_NETWORK socket233 6740 driver5a
SOCKET_NETWORK socket234 6741 driver5b
SOCKET_NETWORK socket235 6742 driver5c
SOCKET_NETWORK socket236 6743 driver5d
SOCKET_NETWORK socket237 6744 driver5a
SOCKET_NETWORK socket238 6745 driver5b
SOCKET_NETWORK socket239 6746 driver5c
SOCKET_NETWORK socket240 6747 driver5d
SOCKET_NETWORK socket241 6748 driver6a
SOCKET_NETWORK socket242 6749 driver6b
SOCKET_NETWORK socket243 6750 driver6c
SOCKET_NETWORK socket244 6751 driver6d
SOCKET_NETWORK socket245 6752 driver6a
SOCKET_NETWORK socket246 6753 driver6b
SOCKET_NETWORK socket247 6754 driver6c
SOCKET_NETWORK socket248 6755 driver6d
SOCKET_NETWORK socket249 6756 driver6a
SOCKET_NETWORK socket250 6757 driver6b
SOCKET_NETWORK socket251 6758 driver6c
SOCKET_NETWORK socket252 6759 driver6d
SOCKET_NETWORK socket253 6760 driver6a
SOCKET_NETWORK socket254 6761 driver6b
SOCKET_NETWORK socket255 6762 driver6c
SOCKET_NETWORK socket256 6763 driver6d
SOCKET_NETWORK socket257 6764 driver6a
SOCKET_NETWORK socket258 6765 driver6b
SOCKET_NETWORK socket259 6766 driver6c
SOCKET_NETWORK socket260 6767 driver6d
SOCKET_NETWORK socket261 6768 driver6a
SOCKET_NETWORK socket262 6769 driver6b
SOCKET_NETWORK socket263 6770 driver6c
SOCKET_NETWORK socket264 6771 driver6d
SOCKET_NETWORK socket265 6772 driver6a
SOCKET_NETWORK socket266 6773 driver6b
SOCKET_NETWORK socket267 6774 driver6c
SOCKET_NETWORK socket268 6775 driver6d
SOCKET_NETWORK socket269 6776 driver6a
SOCKET_NETWORK socket270 6777 driver6b
SOCKET_NETWORK socket271 6778 driver6c
SOCKET_NETWORK socket272 6779 driver6d
SOCKET_NETWORK socket273 6780 driver6a
SOCKET_NETWORK socket274 6781 driver6b
SOCKET_NETWORK socket275 6782 driver6c
SOCKET_NETWORK socket276 6783 driver6d
SOCKET_NETWORK socket277 6784 driver6a
SOCKET_NETWORK socket278 6785 driver6b
SOCKET_NETWORK socket279 6786 driver6c
SOCKET_NETWORK socket280 6787 driver6d
SOCKET_NETWORK socket281 6788 driver6a
SOCKET_NETWORK socket282 6789 driver6b
SOCKET_NETWORK socket283 6790 driver6c
SOCKET_NETWORK socket284 6791 driver6d
SOCKET_NETWORK socket285 6792 driver6a
SOCKET_NETWORK socket286 6793 driver6b
SOCKET_NETWORK socket287 6794 driver6c
SOCKET_NETWORK socket288 6795 driver6d

```
#elif MASTER_NUM2
SOCKET_NETWORK socket289  6700 driver7a
SOCKET_NETWORK socket290  6701 driver7b
SOCKET_NETWORK socket291  6702 driver7c
SOCKET_NETWORK socket292  6703 driver7d
SOCKET_NETWORK socket293  6704 driver7a
SOCKET_NETWORK socket294  6705 driver7b
SOCKET_NETWORK socket295  6706 driver7c
SOCKET_NETWORK socket296  6707 driver7d
SOCKET_NETWORK socket297  6708 driver7a
SOCKET_NETWORK socket298  6709 driver7b
SOCKET_NETWORK socket299  6710 driver7c
SOCKET_NETWORK socket300  6711 driver7d
SOCKET_NETWORK socket301  6712 driver7a
SOCKET_NETWORK socket302  6713 driver7b
SOCKET_NETWORK socket303  6714 driver7c
SOCKET_NETWORK socket304  6715 driver7d
SOCKET_NETWORK socket305  6716 driver7a
SOCKET_NETWORK socket306  6717 driver7b
SOCKET_NETWORK socket307  6718 driver7c
SOCKET_NETWORK socket308  6719 driver7d
SOCKET_NETWORK socket309  6720 driver7a
SOCKET_NETWORK socket310  6721 driver7b
SOCKET_NETWORK socket311  6722 driver7c
SOCKET_NETWORK socket312  6723 driver7d
SOCKET_NETWORK socket313  6724 driver7a
SOCKET_NETWORK socket314  6725 driver7b
SOCKET_NETWORK socket315  6726 driver7c
SOCKET_NETWORK socket316  6727 driver7d
SOCKET_NETWORK socket317  6728 driver7a
SOCKET_NETWORK socket318  6729 driver7b
SOCKET_NETWORK socket319  6730 driver7c
SOCKET_NETWORK socket320  6731 driver7d
SOCKET_NETWORK socket321  6732 driver7a
SOCKET_NETWORK socket322  6733 driver7b
SOCKET_NETWORK socket323  6734 driver7c
SOCKET_NETWORK socket324  6735 driver7d
SOCKET_NETWORK socket325  6736 driver7a
SOCKET_NETWORK socket326  6737 driver7b
SOCKET_NETWORK socket327  6738 driver7c
SOCKET_NETWORK socket328  6739 driver7d
SOCKET_NETWORK socket329  6740 driver7a
SOCKET_NETWORK socket330  6741 driver7b
SOCKET_NETWORK socket331  6742 driver7c
SOCKET_NETWORK socket332  6743 driver7d
SOCKET_NETWORK socket333  6744 driver7a
SOCKET_NETWORK socket334  6745 driver7b
SOCKET_NETWORK socket335  6746 driver7c
SOCKET_NETWORK socket336  6747 driver7d
SOCKET_NETWORK socket337  6748 driver8a
SOCKET_NETWORK socket338  6749 driver8b
SOCKET_NETWORK socket339  6750 driver8c
SOCKET_NETWORK socket340  6751 driver8d
SOCKET_NETWORK socket341  6752 driver8a
SOCKET_NETWORK socket342  6753 driver8b
SOCKET_NETWORK socket343  6754 driver8c
SOCKET_NETWORK socket344  6755 driver8d
SOCKET_NETWORK socket345  6756 driver8a
SOCKET_NETWORK socket346  6757 driver8b
SOCKET_NETWORK socket347  6758 driver8c
SOCKET_NETWORK socket348  6759 driver8d
SOCKET_NETWORK socket349  6760 driver8a
SOCKET_NETWORK socket350  6761 driver8b
SOCKET_NETWORK socket351  6762 driver8c
SOCKET_NETWORK socket352  6763 driver8d
SOCKET_NETWORK socket353  6764 driver8a
SOCKET_NETWORK socket354  6765 driver8b
SOCKET_NETWORK socket355  6766 driver8c
SOCKET_NETWORK socket356  6767 driver8d
SOCKET_NETWORK socket357  6768 driver8a
SOCKET_NETWORK socket358  6769 driver8b
SOCKET_NETWORK socket359  6770 driver8c
SOCKET_NETWORK socket360  6771 driver8d
SOCKET_NETWORK socket361  6772 driver8a
SOCKET_NETWORK socket362  6773 driver8b
SOCKET_NETWORK socket363  6774 driver8c
SOCKET_NETWORK socket364  6775 driver8d
SOCKET_NETWORK socket365  6776 driver8a
SOCKET_NETWORK socket366  6777 driver8b
SOCKET_NETWORK socket367  6778 driver8c
SOCKET_NETWORK socket368  6779 driver8d
SOCKET_NETWORK socket369  6780 driver8a
SOCKET_NETWORK socket370  6781 driver8b
SOCKET_NETWORK socket371  6782 driver8c
SOCKET_NETWORK socket372  6783 driver8d
SOCKET_NETWORK socket373  6784 driver8a
SOCKET_NETWORK socket374  6785 driver8b
SOCKET_NETWORK socket375  6786 driver8c
SOCKET_NETWORK socket376  6787 driver8d
SOCKET_NETWORK socket377  6788 driver8a
SOCKET_NETWORK socket378  6789 driver8b
SOCKET_NETWORK socket379  6790 driver8c
SOCKET_NETWORK socket380  6791 driver8d
SOCKET_NETWORK socket381  6792 driver8a
SOCKET_NETWORK socket382  6793 driver8b
SOCKET_NETWORK socket383  6794 driver8c
SOCKET_NETWORK socket384  6795 driver8d
SOCKET_NETWORK socket385  6700 driver9a
SOCKET_NETWORK socket386  6701 driver9b
SOCKET_NETWORK socket387  6702 driver9c
SOCKET_NETWORK socket388  6703 driver9d
SOCKET_NETWORK socket389  6704 driver9a
SOCKET_NETWORK socket390  6705 driver9b
SOCKET_NETWORK socket391  6706 driver9c
SOCKET_NETWORK socket392  6707 driver9d
SOCKET_NETWORK socket393  6708 driver9a
SOCKET_NETWORK socket394  6709 driver9b
SOCKET_NETWORK socket395  6710 driver9c
SOCKET_NETWORK socket396  6711 driver9d
SOCKET_NETWORK socket397  6712 driver9a
SOCKET_NETWORK socket398  6713 driver9b
SOCKET_NETWORK socket399  6714 driver9c
SOCKET_NETWORK socket400  6715 driver9d
SOCKET_NETWORK socket401  6716 driver9a
SOCKET_NETWORK socket402  6717 driver9b
SOCKET_NETWORK socket403  6718 driver9c
SOCKET_NETWORK socket404  6719 driver9d
SOCKET_NETWORK socket405  6720 driver9a
SOCKET_NETWORK socket406  6721 driver9b
SOCKET_NETWORK socket407  6722 driver9c
SOCKET_NETWORK socket408  6723 driver9d
SOCKET_NETWORK socket409  6724 driver9a
SOCKET_NETWORK socket410  6725 driver9b
SOCKET_NETWORK socket411  6726 driver9c
SOCKET_NETWORK socket412  6727 driver9d
SOCKET_NETWORK socket413  6728 driver9a
SOCKET_NETWORK socket414  6729 driver9b
SOCKET_NETWORK socket415  6730 driver9c
SOCKET_NETWORK socket416  6731 driver9d
SOCKET_NETWORK socket417  6732 driver9a
SOCKET_NETWORK socket418  6733 driver9b
SOCKET_NETWORK socket419  6734 driver9c
SOCKET_NETWORK socket420  6735 driver9d
SOCKET_NETWORK socket421  6736 driver9a
SOCKET_NETWORK socket422  6737 driver9b
SOCKET_NETWORK socket423  6738 driver9c
SOCKET_NETWORK socket424  6739 driver9d
SOCKET_NETWORK socket425  6740 driver9a
SOCKET_NETWORK socket426  6741 driver9b
SOCKET_NETWORK socket427  6742 driver9c
SOCKET_NETWORK socket428  6743 driver9d
SOCKET_NETWORK socket429  6744 driver9a
SOCKET_NETWORK socket430  6745 driver9b
SOCKET_NETWORK socket431  6746 driver9c
SOCKET_NETWORK socket432  6747 driver9d
SOCKET_NETWORK socket433  6748 driver10a
SOCKET_NETWORK socket434  6749 driver10b
SOCKET_NETWORK socket435  6750 driver10c
SOCKET_NETWORK socket436  6751 driver10d
SOCKET_NETWORK socket437  6752 driver10a
SOCKET_NETWORK socket438  6753 driver10b
SOCKET_NETWORK socket439  6754 driver10c
SOCKET_NETWORK socket440  6755 driver10d
SOCKET_NETWORK socket441  6756 driver10a
SOCKET_NETWORK socket442  6757 driver10b
SOCKET_NETWORK socket443  6758 driver10c
SOCKET_NETWORK socket444  6759 driver10d
SOCKET_NETWORK socket445  6760 driver10a
SOCKET_NETWORK socket446  6761 driver10b
SOCKET_NETWORK socket447  6762 driver10c
SOCKET_NETWORK socket448  6763 driver10d
SOCKET_NETWORK socket449  6764 driver10a
SOCKET_NETWORK socket450  6765 driver10b
SOCKET_NETWORK socket451  6766 driver10c
SOCKET_NETWORK socket452  6767 driver10d
SOCKET_NETWORK socket453  6768 driver10a
SOCKET_NETWORK socket454  6769 driver10b
SOCKET_NETWORK socket455  6770 driver10c
SOCKET_NETWORK socket456  6771 driver10d
SOCKET_NETWORK socket457  6772 driver10a
SOCKET_NETWORK socket458  6773 driver10b
SOCKET_NETWORK socket459  6774 driver10c
SOCKET_NETWORK socket460  6775 driver10d
SOCKET_NETWORK socket461  6776 driver10a
SOCKET_NETWORK socket462  6777 driver10b
SOCKET_NETWORK socket463  6778 driver10c
SOCKET_NETWORK socket464  6779 driver10d
SOCKET_NETWORK socket465  6780 driver10a
SOCKET_NETWORK socket466  6781 driver10b
SOCKET_NETWORK socket467  6782 driver10c
SOCKET_NETWORK socket468  6783 driver10d
SOCKET_NETWORK socket469  6784 driver10a
SOCKET_NETWORK socket470  6785 driver10b
SOCKET_NETWORK socket471  6786 driver10c
SOCKET_NETWORK socket472  6787 driver10d
SOCKET_NETWORK socket473  6788 driver10a
SOCKET_NETWORK socket474  6789 driver10b
SOCKET_NETWORK socket475  6790 driver10c
SOCKET_NETWORK socket476  6791 driver10d
SOCKET_NETWORK socket477  6792 driver10a
SOCKET_NETWORK socket478  6793 driver10b
SOCKET_NETWORK socket479  6794 driver10c
SOCKET_NETWORK socket480  6795 driver10d
SOCKET_NETWORK socket481  6700 driver11a
SOCKET_NETWORK socket482  6701 driver11b
SOCKET_NETWORK socket483  6702 driver11c
SOCKET_NETWORK socket484  6703 driver11d
SOCKET_NETWORK socket485  6704 driver11a
```

```
SOCKET_NETWORK socket486  6705 driver11b
SOCKET_NETWORK socket487  6706 driver11c
SOCKET_NETWORK socket488  6707 driver11d
SOCKET_NETWORK socket489  6708 driver11a
SOCKET_NETWORK socket490  6709 driver11b
SOCKET_NETWORK socket491  6710 driver11c
SOCKET_NETWORK socket492  6711 driver11d
SOCKET_NETWORK socket493  6712 driver11a
SOCKET_NETWORK socket494  6713 driver11b
SOCKET_NETWORK socket495  6714 driver11c
SOCKET_NETWORK socket496  6715 driver11d
SOCKET_NETWORK socket497  6716 driver11a
SOCKET_NETWORK socket498  6717 driver11b
SOCKET_NETWORK socket499  6718 driver11c
SOCKET_NETWORK socket500  6719 driver11d
SOCKET_NETWORK socket501  6720 driver11a
SOCKET_NETWORK socket502  6721 driver11b
SOCKET_NETWORK socket503  6722 driver11c
SOCKET_NETWORK socket504  6723 driver11d
SOCKET_NETWORK socket505  6724 driver11a
SOCKET_NETWORK socket506  6725 driver11b
SOCKET_NETWORK socket507  6726 driver11c
SOCKET_NETWORK socket508  6727 driver11d
SOCKET_NETWORK socket509  6728 driver11a
SOCKET_NETWORK socket510  6729 driver11b
SOCKET_NETWORK socket511  6730 driver11c
SOCKET_NETWORK socket512  6731 driver11d
SOCKET_NETWORK socket513  6732 driver11a
SOCKET_NETWORK socket514  6733 driver11b
SOCKET_NETWORK socket515  6734 driver11c
SOCKET_NETWORK socket516  6735 driver11d
SOCKET_NETWORK socket517  6736 driver11a
SOCKET_NETWORK socket518  6737 driver11b
SOCKET_NETWORK socket519  6738 driver11c
SOCKET_NETWORK socket520  6739 driver11d
SOCKET_NETWORK socket521  6740 driver11a
SOCKET_NETWORK socket522  6741 driver11b
SOCKET_NETWORK socket523  6742 driver11c
SOCKET_NETWORK socket524  6743 driver11d
SOCKET_NETWORK socket525  6744 driver11a
SOCKET_NETWORK socket526  6745 driver11b
SOCKET_NETWORK socket527  6746 driver11c
SOCKET_NETWORK socket528  6747 driver11d
SOCKET_NETWORK socket529  6748 driver12a
SOCKET_NETWORK socket530  6749 driver12b
SOCKET_NETWORK socket531  6750 driver12c
SOCKET_NETWORK socket532  6751 driver12d
SOCKET_NETWORK socket533  6752 driver12a
SOCKET_NETWORK socket534  6753 driver12b
SOCKET_NETWORK socket535  6754 driver12c
SOCKET_NETWORK socket536  6755 driver12d
SOCKET_NETWORK socket537  6756 driver12a
SOCKET_NETWORK socket538  6757 driver12b
SOCKET_NETWORK socket539  6758 driver12c
SOCKET_NETWORK socket540  6759 driver12d
SOCKET_NETWORK socket541  6760 driver12a
SOCKET_NETWORK socket542  6761 driver12b
SOCKET_NETWORK socket543  6762 driver12c
SOCKET_NETWORK socket544  6763 driver12d
SOCKET_NETWORK socket545  6764 driver12a
SOCKET_NETWORK socket546  6765 driver12b
SOCKET_NETWORK socket547  6766 driver12c
SOCKET_NETWORK socket548  6767 driver12d
SOCKET_NETWORK socket549  6768 driver12a
SOCKET_NETWORK socket550  6769 driver12b
SOCKET_NETWORK socket551  6770 driver12c
SOCKET_NETWORK socket552  6771 driver12d
SOCKET_NETWORK socket553  6772 driver12a
SOCKET_NETWORK socket554  6773 driver12b
SOCKET_NETWORK socket555  6774 driver12c
SOCKET_NETWORK socket556  6775 driver12d
SOCKET_NETWORK socket557  6776 driver12a
SOCKET_NETWORK socket558  6777 driver12b
SOCKET_NETWORK socket559  6778 driver12c
SOCKET_NETWORK socket560  6779 driver12d
SOCKET_NETWORK socket561  6780 driver12a
SOCKET_NETWORK socket562  6781 driver12b
SOCKET_NETWORK socket563  6782 driver12c
SOCKET_NETWORK socket564  6783 driver12d
SOCKET_NETWORK socket565  6784 driver12a
SOCKET_NETWORK socket566  6785 driver12b
SOCKET_NETWORK socket567  6786 driver12c
SOCKET_NETWORK socket568  6787 driver12d
SOCKET_NETWORK socket569  6788 driver12a
SOCKET_NETWORK socket570  6789 driver12b
SOCKET_NETWORK socket571  6790 driver12c
SOCKET_NETWORK socket572  6791 driver12d
SOCKET_NETWORK socket573  6792 driver12a
SOCKET_NETWORK socket574  6793 driver12b
SOCKET_NETWORK socket575  6794 driver12c
SOCKET_NETWORK socket576  6795 driver12d
#elif MASTER_NUM3
SOCKET_NETWORK socket577  6700 driver13a
SOCKET_NETWORK socket578  6701 driver13b
SOCKET_NETWORK socket579  6702 driver13c
SOCKET_NETWORK socket580  6703 driver13d
SOCKET_NETWORK socket581  6704 driver13a
SOCKET_NETWORK socket582  6705 driver13b
SOCKET_NETWORK socket583  6706 driver13c
SOCKET_NETWORK socket584  6707 driver13d
SOCKET_NETWORK socket585  6708 driver13a
SOCKET_NETWORK socket586  6709 driver13b
SOCKET_NETWORK socket587  6710 driver13c
SOCKET_NETWORK socket588  6711 driver13d
SOCKET_NETWORK socket589  6712 driver13a
SOCKET_NETWORK socket590  6713 driver13b
SOCKET_NETWORK socket591  6714 driver13c
SOCKET_NETWORK socket592  6715 driver13d
SOCKET_NETWORK socket593  6716 driver13a
SOCKET_NETWORK socket594  6717 driver13b
SOCKET_NETWORK socket595  6718 driver13c
SOCKET_NETWORK socket596  6719 driver13d
SOCKET_NETWORK socket597  6720 driver13a
SOCKET_NETWORK socket598  6721 driver13b
SOCKET_NETWORK socket599  6722 driver13c
SOCKET_NETWORK socket600  6723 driver13d
SOCKET_NETWORK socket601  6724 driver13a
SOCKET_NETWORK socket602  6725 driver13b
SOCKET_NETWORK socket603  6726 driver13c
SOCKET_NETWORK socket604  6727 driver13d
SOCKET_NETWORK socket605  6728 driver13a
SOCKET_NETWORK socket606  6729 driver13b
SOCKET_NETWORK socket607  6730 driver13c
SOCKET_NETWORK socket608  6731 driver13d
SOCKET_NETWORK socket609  6732 driver13a
SOCKET_NETWORK socket610  6733 driver13b
SOCKET_NETWORK socket611  6734 driver13c
SOCKET_NETWORK socket612  6735 driver13d
SOCKET_NETWORK socket613  6736 driver13a
SOCKET_NETWORK socket614  6737 driver13b
SOCKET_NETWORK socket615  6738 driver13c
SOCKET_NETWORK socket616  6739 driver13d
SOCKET_NETWORK socket617  6740 driver13a
SOCKET_NETWORK socket618  6741 driver13b
SOCKET_NETWORK socket619  6742 driver13c
SOCKET_NETWORK socket620  6743 driver13d
SOCKET_NETWORK socket621  6744 driver13a
SOCKET_NETWORK socket622  6745 driver13b
SOCKET_NETWORK socket623  6746 driver13c
SOCKET_NETWORK socket624  6747 driver13d
SOCKET_NETWORK socket625  6748 driver14a
SOCKET_NETWORK socket626  6749 driver14b
SOCKET_NETWORK socket627  6750 driver14c
SOCKET_NETWORK socket628  6751 driver14d
SOCKET_NETWORK socket629  6752 driver14a
SOCKET_NETWORK socket630  6753 driver14b
SOCKET_NETWORK socket631  6754 driver14c
SOCKET_NETWORK socket632  6755 driver14d
SOCKET_NETWORK socket633  6756 driver14a
SOCKET_NETWORK socket634  6757 driver14b
SOCKET_NETWORK socket635  6758 driver14c
SOCKET_NETWORK socket636  6759 driver14d
SOCKET_NETWORK socket637  6760 driver14a
SOCKET_NETWORK socket638  6761 driver14b
SOCKET_NETWORK socket639  6762 driver14c
SOCKET_NETWORK socket640  6763 driver14d
SOCKET_NETWORK socket641  6764 driver14a
SOCKET_NETWORK socket642  6765 driver14b
SOCKET_NETWORK socket643  6766 driver14c
SOCKET_NETWORK socket644  6767 driver14d
SOCKET_NETWORK socket645  6768 driver14a
SOCKET_NETWORK socket646  6769 driver14b
SOCKET_NETWORK socket647  6770 driver14c
SOCKET_NETWORK socket648  6771 driver14d
SOCKET_NETWORK socket649  6772 driver14a
SOCKET_NETWORK socket650  6773 driver14b
SOCKET_NETWORK socket651  6774 driver14c
SOCKET_NETWORK socket652  6775 driver14d
SOCKET_NETWORK socket653  6776 driver14a
SOCKET_NETWORK socket654  6777 driver14b
SOCKET_NETWORK socket655  6778 driver14c
SOCKET_NETWORK socket656  6779 driver14d
SOCKET_NETWORK socket657  6780 driver14a
SOCKET_NETWORK socket658  6781 driver14b
SOCKET_NETWORK socket659  6782 driver14c
SOCKET_NETWORK socket660  6783 driver14d
SOCKET_NETWORK socket661  6784 driver14a
SOCKET_NETWORK socket662  6785 driver14b
SOCKET_NETWORK socket663  6786 driver14c
SOCKET_NETWORK socket664  6787 driver14d
SOCKET_NETWORK socket665  6788 driver14a
SOCKET_NETWORK socket666  6789 driver14b
SOCKET_NETWORK socket667  6790 driver14c
SOCKET_NETWORK socket668  6791 driver14d
SOCKET_NETWORK socket669  6792 driver14a
SOCKET_NETWORK socket670  6793 driver14b
SOCKET_NETWORK socket671  6794 driver14c
SOCKET_NETWORK socket672  6795 driver14d
SOCKET_NETWORK socket673  6700 driver15a
SOCKET_NETWORK socket674  6701 driver15b
SOCKET_NETWORK socket675  6702 driver15c
SOCKET_NETWORK socket676  6703 driver15d
SOCKET_NETWORK socket677  6704 driver15a
SOCKET_NETWORK socket678  6705 driver15b
SOCKET_NETWORK socket679  6706 driver15c
SOCKET_NETWORK socket680  6707 driver15d
SOCKET_NETWORK socket681  6708 driver15a
SOCKET_NETWORK socket682  6709 driver15b
```

SOCKET_NETWORK socket683 6710 driver15c
SOCKET_NETWORK socket684 6711 driver15d
SOCKET_NETWORK socket685 6712 driver15a
SOCKET_NETWORK socket686 6713 driver15b
SOCKET_NETWORK socket687 6714 driver15c
SOCKET_NETWORK socket688 6715 driver15d
SOCKET_NETWORK socket689 6716 driver15a
SOCKET_NETWORK socket690 6717 driver15b
SOCKET_NETWORK socket691 6718 driver15c
SOCKET_NETWORK socket692 6719 driver15d
SOCKET_NETWORK socket693 6720 driver15a
SOCKET_NETWORK socket694 6721 driver15b
SOCKET_NETWORK socket695 6722 driver15c
SOCKET_NETWORK socket696 6723 driver15d
SOCKET_NETWORK socket697 6724 driver15a
SOCKET_NETWORK socket698 6725 driver15b
SOCKET_NETWORK socket699 6726 driver15c
SOCKET_NETWORK socket700 6727 driver15d
SOCKET_NETWORK socket701 6728 driver15a
SOCKET_NETWORK socket702 6729 driver15b
SOCKET_NETWORK socket703 6730 driver15c
SOCKET_NETWORK socket704 6731 driver15d
SOCKET_NETWORK socket705 6732 driver15a
SOCKET_NETWORK socket706 6733 driver15b
SOCKET_NETWORK socket707 6734 driver15c
SOCKET_NETWORK socket708 6735 driver15d
SOCKET_NETWORK socket709 6736 driver15a
SOCKET_NETWORK socket710 6737 driver15b
SOCKET_NETWORK socket711 6738 driver15c
SOCKET_NETWORK socket712 6739 driver15d
SOCKET_NETWORK socket713 6740 driver15a
SOCKET_NETWORK socket714 6741 driver15b
SOCKET_NETWORK socket715 6742 driver15c
SOCKET_NETWORK socket716 6743 driver15d
SOCKET_NETWORK socket717 6744 driver15a
SOCKET_NETWORK socket718 6745 driver15b
SOCKET_NETWORK socket719 6746 driver15c
SOCKET_NETWORK socket720 6747 driver15d
SOCKET_NETWORK socket721 6748 driver16a
SOCKET_NETWORK socket722 6749 driver16b
SOCKET_NETWORK socket723 6750 driver16c
SOCKET_NETWORK socket724 6751 driver16d
SOCKET_NETWORK socket725 6752 driver16a
SOCKET_NETWORK socket726 6753 driver16b
SOCKET_NETWORK socket727 6754 driver16c
SOCKET_NETWORK socket728 6755 driver16d
SOCKET_NETWORK socket729 6756 driver16a
SOCKET_NETWORK socket730 6757 driver16b
SOCKET_NETWORK socket731 6758 driver16c
SOCKET_NETWORK socket732 6759 driver16d
SOCKET_NETWORK socket733 6760 driver16a
SOCKET_NETWORK socket734 6761 driver16b
SOCKET_NETWORK socket735 6762 driver16c
SOCKET_NETWORK socket736 6763 driver16d
SOCKET_NETWORK socket737 6764 driver16a
SOCKET_NETWORK socket738 6765 driver16b
SOCKET_NETWORK socket739 6766 driver16c
SOCKET_NETWORK socket740 6767 driver16d
SOCKET_NETWORK socket741 6768 driver16a
SOCKET_NETWORK socket742 6769 driver16b
SOCKET_NETWORK socket743 6770 driver16c
SOCKET_NETWORK socket744 6771 driver16d
SOCKET_NETWORK socket745 6772 driver16a
SOCKET_NETWORK socket746 6773 driver16b
SOCKET_NETWORK socket747 6774 driver16c
SOCKET_NETWORK socket748 6775 driver16d
SOCKET_NETWORK socket749 6776 driver16a
SOCKET_NETWORK socket750 6777 driver16b
SOCKET_NETWORK socket751 6778 driver16c
SOCKET_NETWORK socket752 6779 driver16d
SOCKET_NETWORK socket753 6780 driver16a
SOCKET_NETWORK socket754 6781 driver16b
SOCKET_NETWORK socket755 6782 driver16c
SOCKET_NETWORK socket756 6783 driver16d
SOCKET_NETWORK socket757 6784 driver16a
SOCKET_NETWORK socket758 6785 driver16b
SOCKET_NETWORK socket759 6786 driver16c
SOCKET_NETWORK socket760 6787 driver16d
SOCKET_NETWORK socket761 6788 driver16a
SOCKET_NETWORK socket762 6789 driver16b
SOCKET_NETWORK socket763 6790 driver16c
SOCKET_NETWORK socket764 6791 driver16d
SOCKET_NETWORK socket765 6792 driver16a
SOCKET_NETWORK socket766 6793 driver16b
SOCKET_NETWORK socket767 6794 driver16c
SOCKET_NETWORK socket768 6795 driver16d
SOCKET_NETWORK socket769 6700 driver17a
SOCKET_NETWORK socket770 6701 driver17b
SOCKET_NETWORK socket771 6702 driver17c
SOCKET_NETWORK socket772 6703 driver17d
SOCKET_NETWORK socket773 6704 driver17a
SOCKET_NETWORK socket774 6705 driver17b
SOCKET_NETWORK socket775 6706 driver17c
SOCKET_NETWORK socket776 6707 driver17d
SOCKET_NETWORK socket777 6708 driver17a
SOCKET_NETWORK socket778 6709 driver17b
SOCKET_NETWORK socket779 6710 driver17c
SOCKET_NETWORK socket780 6711 driver17d
SOCKET_NETWORK socket781 6712 driver17a
SOCKET_NETWORK socket782 6713 driver17b
SOCKET_NETWORK socket783 6714 driver17c
SOCKET_NETWORK socket784 6715 driver17d
SOCKET_NETWORK socket785 6716 driver17a
SOCKET_NETWORK socket786 6717 driver17b
SOCKET_NETWORK socket787 6718 driver17c
SOCKET_NETWORK socket788 6719 driver17d
SOCKET_NETWORK socket789 6720 driver17a
SOCKET_NETWORK socket790 6721 driver17b
SOCKET_NETWORK socket791 6722 driver17c
SOCKET_NETWORK socket792 6723 driver17d
SOCKET_NETWORK socket793 6724 driver17a
SOCKET_NETWORK socket794 6725 driver17b
SOCKET_NETWORK socket795 6726 driver17c
SOCKET_NETWORK socket796 6727 driver17d
SOCKET_NETWORK socket797 6728 driver17a
SOCKET_NETWORK socket798 6729 driver17b
SOCKET_NETWORK socket799 6730 driver17c
SOCKET_NETWORK socket800 6731 driver17d
SOCKET_NETWORK socket801 6732 driver17a
SOCKET_NETWORK socket802 6733 driver17b
SOCKET_NETWORK socket803 6734 driver17c
SOCKET_NETWORK socket804 6735 driver17d
SOCKET_NETWORK socket805 6736 driver17a
SOCKET_NETWORK socket806 6737 driver17b
SOCKET_NETWORK socket807 6738 driver17c
SOCKET_NETWORK socket808 6739 driver17d
SOCKET_NETWORK socket809 6740 driver17a
SOCKET_NETWORK socket810 6741 driver17b
SOCKET_NETWORK socket811 6742 driver17c
SOCKET_NETWORK socket812 6743 driver17d
SOCKET_NETWORK socket813 6744 driver17a
SOCKET_NETWORK socket814 6745 driver17b
SOCKET_NETWORK socket815 6746 driver17c
SOCKET_NETWORK socket816 6747 driver17d
SOCKET_NETWORK socket817 6748 driver18a
SOCKET_NETWORK socket818 6749 driver18b
SOCKET_NETWORK socket819 6750 driver18c
SOCKET_NETWORK socket820 6751 driver18d
SOCKET_NETWORK socket821 6752 driver18a
SOCKET_NETWORK socket822 6753 driver18b
SOCKET_NETWORK socket823 6754 driver18c
SOCKET_NETWORK socket824 6755 driver18d
SOCKET_NETWORK socket825 6756 driver18a
SOCKET_NETWORK socket826 6757 driver18b
SOCKET_NETWORK socket827 6758 driver18c
SOCKET_NETWORK socket828 6759 driver18d
SOCKET_NETWORK socket829 6760 driver18a
SOCKET_NETWORK socket830 6761 driver18b
SOCKET_NETWORK socket831 6762 driver18c
SOCKET_NETWORK socket832 6763 driver18d
SOCKET_NETWORK socket833 6764 driver18a
SOCKET_NETWORK socket834 6765 driver18b
SOCKET_NETWORK socket835 6766 driver18c
SOCKET_NETWORK socket836 6767 driver18d
SOCKET_NETWORK socket837 6768 driver18a
SOCKET_NETWORK socket838 6769 driver18b
SOCKET_NETWORK socket839 6770 driver18c
SOCKET_NETWORK socket840 6771 driver18d
SOCKET_NETWORK socket841 6772 driver18a
SOCKET_NETWORK socket842 6773 driver18b
SOCKET_NETWORK socket843 6774 driver18c
SOCKET_NETWORK socket844 6775 driver18d
SOCKET_NETWORK socket845 6776 driver18a
SOCKET_NETWORK socket846 6777 driver18b
SOCKET_NETWORK socket847 6778 driver18c
SOCKET_NETWORK socket848 6779 driver18d
SOCKET_NETWORK socket849 6780 driver18a
SOCKET_NETWORK socket850 6781 driver18b
SOCKET_NETWORK socket851 6782 driver18c
SOCKET_NETWORK socket852 6783 driver18d
SOCKET_NETWORK socket853 6784 driver18a
SOCKET_NETWORK socket854 6785 driver18b
SOCKET_NETWORK socket855 6786 driver18c
SOCKET_NETWORK socket856 6787 driver18d
SOCKET_NETWORK socket857 6788 driver18a
SOCKET_NETWORK socket858 6789 driver18b
SOCKET_NETWORK socket859 6790 driver18c
SOCKET_NETWORK socket860 6791 driver18d
SOCKET_NETWORK socket861 6792 driver18a
SOCKET_NETWORK socket862 6793 driver18b
SOCKET_NETWORK socket863 6794 driver18c
SOCKET_NETWORK socket864 6795 driver18d
#elif MASTER_NUM4
SOCKET_NETWORK socket865 6700 driver19a
SOCKET_NETWORK socket866 6701 driver19b
SOCKET_NETWORK socket867 6702 driver19c
SOCKET_NETWORK socket868 6703 driver19d
SOCKET_NETWORK socket869 6704 driver19a
SOCKET_NETWORK socket870 6705 driver19b
SOCKET_NETWORK socket871 6706 driver19c
SOCKET_NETWORK socket872 6707 driver19d
SOCKET_NETWORK socket873 6708 driver19a
SOCKET_NETWORK socket874 6709 driver19b
SOCKET_NETWORK socket875 6710 driver19c
SOCKET_NETWORK socket876 6711 driver19d
SOCKET_NETWORK socket877 6712 driver19a
SOCKET_NETWORK socket878 6713 driver19b
SOCKET_NETWORK socket879 6714 driver19c

SOCKET_NETWORK socket880 6715 driver19d
SOCKET_NETWORK socket881 6716 driver19a
SOCKET_NETWORK socket882 6717 driver19b
SOCKET_NETWORK socket883 6718 driver19c
SOCKET_NETWORK socket884 6719 driver19d
SOCKET_NETWORK socket885 6720 driver19a
SOCKET_NETWORK socket886 6721 driver19b
SOCKET_NETWORK socket887 6722 driver19c
SOCKET_NETWORK socket888 6723 driver19d
SOCKET_NETWORK socket889 6724 driver19a
SOCKET_NETWORK socket890 6725 driver19b
SOCKET_NETWORK socket891 6726 driver19c
SOCKET_NETWORK socket892 6727 driver19d
SOCKET_NETWORK socket893 6728 driver19a
SOCKET_NETWORK socket894 6729 driver19b
SOCKET_NETWORK socket895 6730 driver19c
SOCKET_NETWORK socket896 6731 driver19d
SOCKET_NETWORK socket897 6732 driver19a
SOCKET_NETWORK socket898 6733 driver19b
SOCKET_NETWORK socket899 6734 driver19c
SOCKET_NETWORK socket900 6735 driver19d
SOCKET_NETWORK socket901 6736 driver19a
SOCKET_NETWORK socket902 6737 driver19b
SOCKET_NETWORK socket903 6738 driver19c
SOCKET_NETWORK socket904 6739 driver19d
SOCKET_NETWORK socket905 6740 driver19a
SOCKET_NETWORK socket906 6741 driver19b
SOCKET_NETWORK socket907 6742 driver19c
SOCKET_NETWORK socket908 6743 driver19d
SOCKET_NETWORK socket909 6744 driver19a
SOCKET_NETWORK socket910 6745 driver19b
SOCKET_NETWORK socket911 6746 driver19c
SOCKET_NETWORK socket912 6747 driver19d
SOCKET_NETWORK socket913 6748 driver20a
SOCKET_NETWORK socket914 6749 driver20b
SOCKET_NETWORK socket915 6750 driver20c
SOCKET_NETWORK socket916 6751 driver20d
SOCKET_NETWORK socket917 6752 driver20a
SOCKET_NETWORK socket918 6753 driver20b
SOCKET_NETWORK socket919 6754 driver20c
SOCKET_NETWORK socket920 6755 driver20d
SOCKET_NETWORK socket921 6756 driver20a
SOCKET_NETWORK socket922 6757 driver20b
SOCKET_NETWORK socket923 6758 driver20c
SOCKET_NETWORK socket924 6759 driver20d
SOCKET_NETWORK socket925 6760 driver20a
SOCKET_NETWORK socket926 6761 driver20b
SOCKET_NETWORK socket927 6762 driver20c
SOCKET_NETWORK socket928 6763 driver20d
SOCKET_NETWORK socket929 6764 driver20a
SOCKET_NETWORK socket930 6765 driver20b
SOCKET_NETWORK socket931 6766 driver20c
SOCKET_NETWORK socket932 6767 driver20d
SOCKET_NETWORK socket933 6768 driver20a
SOCKET_NETWORK socket934 6769 driver20b
SOCKET_NETWORK socket935 6770 driver20c
SOCKET_NETWORK socket936 6771 driver20d
SOCKET_NETWORK socket937 6772 driver20a
SOCKET_NETWORK socket938 6773 driver20b
SOCKET_NETWORK socket939 6774 driver20c
SOCKET_NETWORK socket940 6775 driver20d
SOCKET_NETWORK socket941 6776 driver20a
SOCKET_NETWORK socket942 6777 driver20b
SOCKET_NETWORK socket943 6778 driver20c
SOCKET_NETWORK socket944 6779 driver20d
SOCKET_NETWORK socket945 6780 driver20a
SOCKET_NETWORK socket946 6781 driver20b
SOCKET_NETWORK socket947 6782 driver20c
SOCKET_NETWORK socket948 6783 driver20d
SOCKET_NETWORK socket949 6784 driver20a
SOCKET_NETWORK socket950 6785 driver20b
SOCKET_NETWORK socket951 6786 driver20c
SOCKET_NETWORK socket952 6787 driver20d
SOCKET_NETWORK socket953 6788 driver20a
SOCKET_NETWORK socket954 6789 driver20b
SOCKET_NETWORK socket955 6790 driver20c
SOCKET_NETWORK socket956 6791 driver20d
SOCKET_NETWORK socket957 6792 driver20a
SOCKET_NETWORK socket958 6793 driver20b
SOCKET_NETWORK socket959 6794 driver20c
SOCKET_NETWORK socket960 6795 driver20d
SOCKET_NETWORK socket961 6700 driver21a
SOCKET_NETWORK socket962 6701 driver21b
SOCKET_NETWORK socket963 6702 driver21c
SOCKET_NETWORK socket964 6703 driver21d
SOCKET_NETWORK socket965 6704 driver21a
SOCKET_NETWORK socket966 6705 driver21b
SOCKET_NETWORK socket967 6706 driver21c
SOCKET_NETWORK socket968 6707 driver21d
SOCKET_NETWORK socket969 6708 driver21a
SOCKET_NETWORK socket970 6709 driver21b
SOCKET_NETWORK socket971 6710 driver21c
SOCKET_NETWORK socket972 6711 driver21d
SOCKET_NETWORK socket973 6712 driver21a
SOCKET_NETWORK socket974 6713 driver21b
SOCKET_NETWORK socket975 6714 driver21c
SOCKET_NETWORK socket976 6715 driver21d
SOCKET_NETWORK socket977 6716 driver21a
SOCKET_NETWORK socket978 6717 driver21b
SOCKET_NETWORK socket979 6718 driver21c
SOCKET_NETWORK socket980 6719 driver21d
SOCKET_NETWORK socket981 6720 driver21a
SOCKET_NETWORK socket982 6721 driver21b
SOCKET_NETWORK socket983 6722 driver21c
SOCKET_NETWORK socket984 6723 driver21d
SOCKET_NETWORK socket985 6724 driver21a
SOCKET_NETWORK socket986 6725 driver21b
SOCKET_NETWORK socket987 6726 driver21c
SOCKET_NETWORK socket988 6727 driver21d
SOCKET_NETWORK socket989 6728 driver21a
SOCKET_NETWORK socket990 6729 driver21b
SOCKET_NETWORK socket991 6730 driver21c
SOCKET_NETWORK socket992 6731 driver21d
SOCKET_NETWORK socket993 6732 driver21a
SOCKET_NETWORK socket994 6733 driver21b
SOCKET_NETWORK socket995 6734 driver21c
SOCKET_NETWORK socket996 6735 driver21d
SOCKET_NETWORK socket997 6736 driver21a
SOCKET_NETWORK socket998 6737 driver21b
SOCKET_NETWORK socket999 6738 driver21c
SOCKET_NETWORK socket1000 6739 driver21d
SOCKET_NETWORK socket1001 6740 driver21a
SOCKET_NETWORK socket1002 6741 driver21b
SOCKET_NETWORK socket1003 6742 driver21c
SOCKET_NETWORK socket1004 6743 driver21d
SOCKET_NETWORK socket1005 6744 driver21a
SOCKET_NETWORK socket1006 6745 driver21b
SOCKET_NETWORK socket1007 6746 driver21c
SOCKET_NETWORK socket1008 6747 driver21d
SOCKET_NETWORK socket1009 6748 driver22a
SOCKET_NETWORK socket1010 6749 driver22b
SOCKET_NETWORK socket1011 6750 driver22c
SOCKET_NETWORK socket1012 6751 driver22d
SOCKET_NETWORK socket1013 6752 driver22a
SOCKET_NETWORK socket1014 6753 driver22b
SOCKET_NETWORK socket1015 6754 driver22c
SOCKET_NETWORK socket1016 6755 driver22d
SOCKET_NETWORK socket1017 6756 driver22a
SOCKET_NETWORK socket1018 6757 driver22b
SOCKET_NETWORK socket1019 6758 driver22c
SOCKET_NETWORK socket1020 6759 driver22d
SOCKET_NETWORK socket1021 6760 driver22a
SOCKET_NETWORK socket1022 6761 driver22b
SOCKET_NETWORK socket1023 6762 driver22c
SOCKET_NETWORK socket1024 6763 driver22d
SOCKET_NETWORK socket1025 6764 driver22a
SOCKET_NETWORK socket1026 6765 driver22b
SOCKET_NETWORK socket1027 6766 driver22c
SOCKET_NETWORK socket1028 6767 driver22d
SOCKET_NETWORK socket1029 6768 driver22a
SOCKET_NETWORK socket1030 6769 driver22b
SOCKET_NETWORK socket1031 6770 driver22c
SOCKET_NETWORK socket1032 6771 driver22d
SOCKET_NETWORK socket1033 6772 driver22a
SOCKET_NETWORK socket1034 6773 driver22b
SOCKET_NETWORK socket1035 6774 driver22c
SOCKET_NETWORK socket1036 6775 driver22d
SOCKET_NETWORK socket1037 6776 driver22a
SOCKET_NETWORK socket1038 6777 driver22b
SOCKET_NETWORK socket1039 6778 driver22c
SOCKET_NETWORK socket1040 6779 driver22d
SOCKET_NETWORK socket1041 6780 driver22a
SOCKET_NETWORK socket1042 6781 driver22b
SOCKET_NETWORK socket1043 6782 driver22c
SOCKET_NETWORK socket1044 6783 driver22d
SOCKET_NETWORK socket1045 6784 driver22a
SOCKET_NETWORK socket1046 6785 driver22b
SOCKET_NETWORK socket1047 6786 driver22c
SOCKET_NETWORK socket1048 6787 driver22d
SOCKET_NETWORK socket1049 6788 driver22a
SOCKET_NETWORK socket1050 6789 driver22b
SOCKET_NETWORK socket1051 6790 driver22c
SOCKET_NETWORK socket1052 6791 driver22d
SOCKET_NETWORK socket1053 6792 driver22a
SOCKET_NETWORK socket1054 6793 driver22b
SOCKET_NETWORK socket1055 6794 driver22c
SOCKET_NETWORK socket1056 6795 driver22d
SOCKET_NETWORK socket1057 6700 driver23a
SOCKET_NETWORK socket1058 6701 driver23b
SOCKET_NETWORK socket1059 6702 driver23c
SOCKET_NETWORK socket1060 6703 driver23d
SOCKET_NETWORK socket1061 6704 driver23a
SOCKET_NETWORK socket1062 6705 driver23b
SOCKET_NETWORK socket1063 6706 driver23c
SOCKET_NETWORK socket1064 6707 driver23d
SOCKET_NETWORK socket1065 6708 driver23a
SOCKET_NETWORK socket1066 6709 driver23b
SOCKET_NETWORK socket1067 6710 driver23c
SOCKET_NETWORK socket1068 6711 driver23d
SOCKET_NETWORK socket1069 6712 driver23a
SOCKET_NETWORK socket1070 6713 driver23b
SOCKET_NETWORK socket1071 6714 driver23c
SOCKET_NETWORK socket1072 6715 driver23d
SOCKET_NETWORK socket1073 6716 driver23a
SOCKET_NETWORK socket1074 6717 driver23b
SOCKET_NETWORK socket1075 6718 driver23c
SOCKET_NETWORK socket1076 6719 driver23d
SOCKET_NETWORK socket1077 6720 driver23a

SOCKET_NETWORK socket1078 6721 driver23b
SOCKET_NETWORK socket1079 6722 driver23c
SOCKET_NETWORK socket1080 6723 driver23d
SOCKET_NETWORK socket1081 6724 driver23a
SOCKET_NETWORK socket1082 6725 driver23b
SOCKET_NETWORK socket1083 6726 driver23c
SOCKET_NETWORK socket1084 6727 driver23d
SOCKET_NETWORK socket1085 6728 driver23a
SOCKET_NETWORK socket1086 6729 driver23b
SOCKET_NETWORK socket1087 6730 driver23c
SOCKET_NETWORK socket1088 6731 driver23d
SOCKET_NETWORK socket1089 6732 driver23a
SOCKET_NETWORK socket1090 6733 driver23b
SOCKET_NETWORK socket1091 6734 driver23c
SOCKET_NETWORK socket1092 6735 driver23d
SOCKET_NETWORK socket1093 6736 driver23a
SOCKET_NETWORK socket1094 6737 driver23b
SOCKET_NETWORK socket1095 6738 driver23c
SOCKET_NETWORK socket1096 6739 driver23d
SOCKET_NETWORK socket1097 6740 driver23a
SOCKET_NETWORK socket1098 6741 driver23b
SOCKET_NETWORK socket1099 6742 driver23c
SOCKET_NETWORK socket1100 6743 driver23d
SOCKET_NETWORK socket1101 6744 driver23a
SOCKET_NETWORK socket1102 6745 driver23b
SOCKET_NETWORK socket1103 6746 driver23c
SOCKET_NETWORK socket1104 6747 driver23d
SOCKET_NETWORK socket1105 6748 driver24a
SOCKET_NETWORK socket1106 6749 driver24b
SOCKET_NETWORK socket1107 6750 driver24c
SOCKET_NETWORK socket1108 6751 driver24d
SOCKET_NETWORK socket1109 6752 driver24a
SOCKET_NETWORK socket1110 6753 driver24b
SOCKET_NETWORK socket1111 6754 driver24c
SOCKET_NETWORK socket1112 6755 driver24d
SOCKET_NETWORK socket1113 6756 driver24a
SOCKET_NETWORK socket1114 6757 driver24b
SOCKET_NETWORK socket1115 6758 driver24c
SOCKET_NETWORK socket1116 6759 driver24d
SOCKET_NETWORK socket1117 6760 driver24a
SOCKET_NETWORK socket1118 6761 driver24b
SOCKET_NETWORK socket1119 6762 driver24c
SOCKET_NETWORK socket1120 6763 driver24d
SOCKET_NETWORK socket1121 6764 driver24a
SOCKET_NETWORK socket1122 6765 driver24b
SOCKET_NETWORK socket1123 6766 driver24c
SOCKET_NETWORK socket1124 6767 driver24d
SOCKET_NETWORK socket1125 6768 driver24a
SOCKET_NETWORK socket1126 6769 driver24b
SOCKET_NETWORK socket1127 6770 driver24c
SOCKET_NETWORK socket1128 6771 driver24d
SOCKET_NETWORK socket1129 6772 driver24a
SOCKET_NETWORK socket1130 6773 driver24b
SOCKET_NETWORK socket1131 6774 driver24c
SOCKET_NETWORK socket1132 6775 driver24d
SOCKET_NETWORK socket1133 6776 driver24a
SOCKET_NETWORK socket1134 6777 driver24b
SOCKET_NETWORK socket1135 6778 driver24c
SOCKET_NETWORK socket1136 6779 driver24d
SOCKET_NETWORK socket1137 6780 driver24a
SOCKET_NETWORK socket1138 6781 driver24b
SOCKET_NETWORK socket1139 6782 driver24c
SOCKET_NETWORK socket1140 6783 driver24d
SOCKET_NETWORK socket1141 6784 driver24a
SOCKET_NETWORK socket1142 6785 driver24b
SOCKET_NETWORK socket1143 6786 driver24c
SOCKET_NETWORK socket1144 6787 driver24d
SOCKET_NETWORK socket1145 6788 driver24a
SOCKET_NETWORK socket1146 6789 driver24b
SOCKET_NETWORK socket1147 6790 driver24c
SOCKET_NETWORK socket1148 6791 driver24d
SOCKET_NETWORK socket1149 6792 driver24a
SOCKET_NETWORK socket1150 6793 driver24b
SOCKET_NETWORK socket1151 6794 driver24c
SOCKET_NETWORK socket1152 6795 driver24d
#elif MASTER_NUM5
SOCKET_NETWORK socket1153 6700 driver25a
SOCKET_NETWORK socket1154 6701 driver25b
SOCKET_NETWORK socket1155 6702 driver25c
SOCKET_NETWORK socket1156 6703 driver25d
SOCKET_NETWORK socket1157 6704 driver25a
SOCKET_NETWORK socket1158 6705 driver25b
SOCKET_NETWORK socket1159 6706 driver25c
SOCKET_NETWORK socket1160 6707 driver25d
SOCKET_NETWORK socket1161 6708 driver25a
SOCKET_NETWORK socket1162 6709 driver25b
SOCKET_NETWORK socket1163 6710 driver25c
SOCKET_NETWORK socket1164 6711 driver25d
SOCKET_NETWORK socket1165 6712 driver25a
SOCKET_NETWORK socket1166 6713 driver25b
SOCKET_NETWORK socket1167 6714 driver25c
SOCKET_NETWORK socket1168 6715 driver25d
SOCKET_NETWORK socket1169 6716 driver25a
SOCKET_NETWORK socket1170 6717 driver25b
SOCKET_NETWORK socket1171 6718 driver25c
SOCKET_NETWORK socket1172 6719 driver25d
SOCKET_NETWORK socket1173 6720 driver25a
SOCKET_NETWORK socket1174 6721 driver25b
SOCKET_NETWORK socket1175 6722 driver25c
SOCKET_NETWORK socket1176 6723 driver25d
SOCKET_NETWORK socket1177 6724 driver25a
SOCKET_NETWORK socket1178 6725 driver25b
SOCKET_NETWORK socket1179 6726 driver25c
SOCKET_NETWORK socket1180 6727 driver25d
SOCKET_NETWORK socket1181 6728 driver25a
SOCKET_NETWORK socket1182 6729 driver25b
SOCKET_NETWORK socket1183 6730 driver25c
SOCKET_NETWORK socket1184 6731 driver25d
SOCKET_NETWORK socket1185 6732 driver25a
SOCKET_NETWORK socket1186 6733 driver25b
SOCKET_NETWORK socket1187 6734 driver25c
SOCKET_NETWORK socket1188 6735 driver25d
SOCKET_NETWORK socket1189 6736 driver25a
SOCKET_NETWORK socket1190 6737 driver25b
SOCKET_NETWORK socket1191 6738 driver25c
SOCKET_NETWORK socket1192 6739 driver25d
SOCKET_NETWORK socket1193 6740 driver25a
SOCKET_NETWORK socket1194 6741 driver25b
SOCKET_NETWORK socket1195 6742 driver25c
SOCKET_NETWORK socket1196 6743 driver25d
SOCKET_NETWORK socket1197 6744 driver25a
SOCKET_NETWORK socket1198 6745 driver25b
SOCKET_NETWORK socket1199 6746 driver25c
SOCKET_NETWORK socket1200 6747 driver25d
SOCKET_NETWORK socket1201 6748 driver26a
SOCKET_NETWORK socket1202 6749 driver26b
SOCKET_NETWORK socket1203 6750 driver26c
SOCKET_NETWORK socket1204 6751 driver26d
SOCKET_NETWORK socket1205 6752 driver26a
SOCKET_NETWORK socket1206 6753 driver26b
SOCKET_NETWORK socket1207 6754 driver26c
SOCKET_NETWORK socket1208 6755 driver26d
SOCKET_NETWORK socket1209 6756 driver26a
SOCKET_NETWORK socket1210 6757 driver26b
SOCKET_NETWORK socket1211 6758 driver26c
SOCKET_NETWORK socket1212 6759 driver26d
SOCKET_NETWORK socket1213 6760 driver26a
SOCKET_NETWORK socket1214 6761 driver26b
SOCKET_NETWORK socket1215 6762 driver26c
SOCKET_NETWORK socket1216 6763 driver26d
SOCKET_NETWORK socket1217 6764 driver26a
SOCKET_NETWORK socket1218 6765 driver26b
SOCKET_NETWORK socket1219 6766 driver26c
SOCKET_NETWORK socket1220 6767 driver26d
SOCKET_NETWORK socket1221 6768 driver26a
SOCKET_NETWORK socket1222 6769 driver26b
SOCKET_NETWORK socket1223 6770 driver26c
SOCKET_NETWORK socket1224 6771 driver26d
SOCKET_NETWORK socket1225 6772 driver26a
SOCKET_NETWORK socket1226 6773 driver26b
SOCKET_NETWORK socket1227 6774 driver26c
SOCKET_NETWORK socket1228 6775 driver26d
SOCKET_NETWORK socket1229 6776 driver26a
SOCKET_NETWORK socket1230 6777 driver26b
SOCKET_NETWORK socket1231 6778 driver26c
SOCKET_NETWORK socket1232 6779 driver26d
SOCKET_NETWORK socket1233 6780 driver26a
SOCKET_NETWORK socket1234 6781 driver26b
SOCKET_NETWORK socket1235 6782 driver26c
SOCKET_NETWORK socket1236 6783 driver26d
SOCKET_NETWORK socket1237 6784 driver26a
SOCKET_NETWORK socket1238 6785 driver26b
SOCKET_NETWORK socket1239 6786 driver26c
SOCKET_NETWORK socket1240 6787 driver26d
SOCKET_NETWORK socket1241 6788 driver26a
SOCKET_NETWORK socket1242 6789 driver26b
SOCKET_NETWORK socket1243 6790 driver26c
SOCKET_NETWORK socket1244 6791 driver26d
SOCKET_NETWORK socket1245 6792 driver26a
SOCKET_NETWORK socket1246 6793 driver26b
SOCKET_NETWORK socket1247 6794 driver26c
SOCKET_NETWORK socket1248 6795 driver26d
SOCKET_NETWORK socket1249 6700 driver27a
SOCKET_NETWORK socket1250 6701 driver27b
SOCKET_NETWORK socket1251 6702 driver27c
SOCKET_NETWORK socket1252 6703 driver27d
SOCKET_NETWORK socket1253 6704 driver27a
SOCKET_NETWORK socket1254 6705 driver27b
SOCKET_NETWORK socket1255 6706 driver27c
SOCKET_NETWORK socket1256 6707 driver27d
SOCKET_NETWORK socket1257 6708 driver27a
SOCKET_NETWORK socket1258 6709 driver27b
SOCKET_NETWORK socket1259 6710 driver27c
SOCKET_NETWORK socket1260 6711 driver27d
SOCKET_NETWORK socket1261 6712 driver27a
SOCKET_NETWORK socket1262 6713 driver27b
SOCKET_NETWORK socket1263 6714 driver27c
SOCKET_NETWORK socket1264 6715 driver27d
SOCKET_NETWORK socket1265 6716 driver27a
SOCKET_NETWORK socket1266 6717 driver27b
SOCKET_NETWORK socket1267 6718 driver27c
SOCKET_NETWORK socket1268 6719 driver27d
SOCKET_NETWORK socket1269 6720 driver27a
SOCKET_NETWORK socket1270 6721 driver27b
SOCKET_NETWORK socket1271 6722 driver27c
SOCKET_NETWORK socket1272 6723 driver27d
SOCKET_NETWORK socket1273 6724 driver27a
SOCKET_NETWORK socket1274 6725 driver27b

SOCKET_NETWORK socket1275 6726 driver27c
SOCKET_NETWORK socket1276 6727 driver27d
SOCKET_NETWORK socket1277 6728 driver27a
SOCKET_NETWORK socket1278 6729 driver27b
SOCKET_NETWORK socket1279 6730 driver27c
SOCKET_NETWORK socket1280 6731 driver27d
SOCKET_NETWORK socket1281 6732 driver27a
SOCKET_NETWORK socket1282 6733 driver27b
SOCKET_NETWORK socket1283 6734 driver27c
SOCKET_NETWORK socket1284 6735 driver27d
SOCKET_NETWORK socket1285 6736 driver27a
SOCKET_NETWORK socket1286 6737 driver27b
SOCKET_NETWORK socket1287 6738 driver27c
SOCKET_NETWORK socket1288 6739 driver27d
SOCKET_NETWORK socket1289 6740 driver27a
SOCKET_NETWORK socket1290 6741 driver27b
SOCKET_NETWORK socket1291 6742 driver27c
SOCKET_NETWORK socket1292 6743 driver27d
SOCKET_NETWORK socket1293 6744 driver27a
SOCKET_NETWORK socket1294 6745 driver27b
SOCKET_NETWORK socket1295 6746 driver27c
SOCKET_NETWORK socket1296 6747 driver27d
SOCKET_NETWORK socket1297 6748 driver28a
SOCKET_NETWORK socket1298 6749 driver28b
SOCKET_NETWORK socket1299 6750 driver28c
SOCKET_NETWORK socket1300 6751 driver28d
SOCKET_NETWORK socket1301 6752 driver28a
SOCKET_NETWORK socket1302 6753 driver28b
SOCKET_NETWORK socket1303 6754 driver28c
SOCKET_NETWORK socket1304 6755 driver28d
SOCKET_NETWORK socket1305 6756 driver28a
SOCKET_NETWORK socket1306 6757 driver28b
SOCKET_NETWORK socket1307 6758 driver28c
SOCKET_NETWORK socket1308 6759 driver28d
SOCKET_NETWORK socket1309 6760 driver28a
SOCKET_NETWORK socket1310 6761 driver28b
SOCKET_NETWORK socket1311 6762 driver28c
SOCKET_NETWORK socket1312 6763 driver28d
SOCKET_NETWORK socket1313 6764 driver28a
SOCKET_NETWORK socket1314 6765 driver28b
SOCKET_NETWORK socket1315 6766 driver28c
SOCKET_NETWORK socket1316 6767 driver28d
SOCKET_NETWORK socket1317 6768 driver28a
SOCKET_NETWORK socket1318 6769 driver28b
SOCKET_NETWORK socket1319 6770 driver28c
SOCKET_NETWORK socket1320 6771 driver28d
SOCKET_NETWORK socket1321 6772 driver28a
SOCKET_NETWORK socket1322 6773 driver28b
SOCKET_NETWORK socket1323 6774 driver28c
SOCKET_NETWORK socket1324 6775 driver28d
SOCKET_NETWORK socket1325 6776 driver28a
SOCKET_NETWORK socket1326 6777 driver28b
SOCKET_NETWORK socket1327 6778 driver28c
SOCKET_NETWORK socket1328 6779 driver28d
SOCKET_NETWORK socket1329 6780 driver28a
SOCKET_NETWORK socket1330 6781 driver28b
SOCKET_NETWORK socket1331 6782 driver28c
SOCKET_NETWORK socket1332 6783 driver28d
SOCKET_NETWORK socket1333 6784 driver28a
SOCKET_NETWORK socket1334 6785 driver28b
SOCKET_NETWORK socket1335 6786 driver28c
SOCKET_NETWORK socket1336 6787 driver28d
SOCKET_NETWORK socket1337 6788 driver28a
SOCKET_NETWORK socket1338 6789 driver28b
SOCKET_NETWORK socket1339 6790 driver28c
SOCKET_NETWORK socket1340 6791 driver28d
SOCKET_NETWORK socket1341 6792 driver28a
SOCKET_NETWORK socket1342 6793 driver28b
SOCKET_NETWORK socket1343 6794 driver28c
SOCKET_NETWORK socket1344 6795 driver28d
SOCKET_NETWORK socket1345 6700 driver29a
SOCKET_NETWORK socket1346 6701 driver29b
SOCKET_NETWORK socket1347 6702 driver29c
SOCKET_NETWORK socket1348 6703 driver29d
SOCKET_NETWORK socket1349 6704 driver29a
SOCKET_NETWORK socket1350 6705 driver29b
SOCKET_NETWORK socket1351 6706 driver29c
SOCKET_NETWORK socket1352 6707 driver29d
SOCKET_NETWORK socket1353 6708 driver29a
SOCKET_NETWORK socket1354 6709 driver29b
SOCKET_NETWORK socket1355 6710 driver29c
SOCKET_NETWORK socket1356 6711 driver29d
SOCKET_NETWORK socket1357 6712 driver29a
SOCKET_NETWORK socket1358 6713 driver29b
SOCKET_NETWORK socket1359 6714 driver29c
SOCKET_NETWORK socket1360 6715 driver29d
SOCKET_NETWORK socket1361 6716 driver29a
SOCKET_NETWORK socket1362 6717 driver29b
SOCKET_NETWORK socket1363 6718 driver29c
SOCKET_NETWORK socket1364 6719 driver29d
SOCKET_NETWORK socket1365 6720 driver29a
SOCKET_NETWORK socket1366 6721 driver29b
SOCKET_NETWORK socket1367 6722 driver29c
SOCKET_NETWORK socket1368 6723 driver29d
SOCKET_NETWORK socket1369 6724 driver29a
SOCKET_NETWORK socket1370 6725 driver29b
SOCKET_NETWORK socket1371 6726 driver29c
SOCKET_NETWORK socket1372 6727 driver29d
SOCKET_NETWORK socket1373 6728 driver29a
SOCKET_NETWORK socket1374 6729 driver29b
SOCKET_NETWORK socket1375 6730 driver29c
SOCKET_NETWORK socket1376 6731 driver29d
SOCKET_NETWORK socket1377 6732 driver29a
SOCKET_NETWORK socket1378 6733 driver29b
SOCKET_NETWORK socket1379 6734 driver29c
SOCKET_NETWORK socket1380 6735 driver29d
SOCKET_NETWORK socket1381 6736 driver29a
SOCKET_NETWORK socket1382 6737 driver29b
SOCKET_NETWORK socket1383 6738 driver29c
SOCKET_NETWORK socket1384 6739 driver29d
SOCKET_NETWORK socket1385 6740 driver29a
SOCKET_NETWORK socket1386 6741 driver29b
SOCKET_NETWORK socket1387 6742 driver29c
SOCKET_NETWORK socket1388 6743 driver29d
SOCKET_NETWORK socket1389 6744 driver29a
SOCKET_NETWORK socket1390 6745 driver29b
SOCKET_NETWORK socket1391 6746 driver29c
SOCKET_NETWORK socket1392 6747 driver29d
SOCKET_NETWORK socket1393 6748 driver30a
SOCKET_NETWORK socket1394 6749 driver30b
SOCKET_NETWORK socket1395 6750 driver30c
SOCKET_NETWORK socket1396 6751 driver30d
SOCKET_NETWORK socket1397 6752 driver30a
SOCKET_NETWORK socket1398 6753 driver30b
SOCKET_NETWORK socket1399 6754 driver30c
SOCKET_NETWORK socket1400 6755 driver30d
SOCKET_NETWORK socket1401 6756 driver30a
SOCKET_NETWORK socket1402 6757 driver30b
SOCKET_NETWORK socket1403 6758 driver30c
SOCKET_NETWORK socket1404 6759 driver30d
SOCKET_NETWORK socket1405 6760 driver30a
SOCKET_NETWORK socket1406 6761 driver30b
SOCKET_NETWORK socket1407 6762 driver30c
SOCKET_NETWORK socket1408 6763 driver30d
SOCKET_NETWORK socket1409 6764 driver30a
SOCKET_NETWORK socket1410 6765 driver30b
SOCKET_NETWORK socket1411 6766 driver30c
SOCKET_NETWORK socket1412 6767 driver30d
SOCKET_NETWORK socket1413 6768 driver30a
SOCKET_NETWORK socket1414 6769 driver30b
SOCKET_NETWORK socket1415 6770 driver30c
SOCKET_NETWORK socket1416 6771 driver30d
SOCKET_NETWORK socket1417 6772 driver30a
SOCKET_NETWORK socket1418 6773 driver30b
SOCKET_NETWORK socket1419 6774 driver30c
SOCKET_NETWORK socket1420 6775 driver30d
SOCKET_NETWORK socket1421 6776 driver30a
SOCKET_NETWORK socket1422 6777 driver30b
SOCKET_NETWORK socket1423 6778 driver30c
SOCKET_NETWORK socket1424 6779 driver30d
SOCKET_NETWORK socket1425 6780 driver30a
SOCKET_NETWORK socket1426 6781 driver30b
SOCKET_NETWORK socket1427 6782 driver30c
SOCKET_NETWORK socket1428 6783 driver30d
SOCKET_NETWORK socket1429 6784 driver30a
SOCKET_NETWORK socket1430 6785 driver30b
SOCKET_NETWORK socket1431 6786 driver30c
SOCKET_NETWORK socket1432 6787 driver30d
SOCKET_NETWORK socket1433 6788 driver30a
SOCKET_NETWORK socket1434 6789 driver30b
SOCKET_NETWORK socket1435 6790 driver30c
SOCKET_NETWORK socket1436 6791 driver30d
SOCKET_NETWORK socket1437 6792 driver30a
SOCKET_NETWORK socket1438 6793 driver30b
SOCKET_NETWORK socket1439 6794 driver30c
SOCKET_NETWORK socket1440 6795 driver30d
#elif MASTER_NUM6
SOCKET_NETWORK socket1441 6700 driver31a
SOCKET_NETWORK socket1442 6701 driver31b
SOCKET_NETWORK socket1443 6702 driver31c
SOCKET_NETWORK socket1444 6703 driver31d
SOCKET_NETWORK socket1445 6704 driver31a
SOCKET_NETWORK socket1446 6705 driver31b
SOCKET_NETWORK socket1447 6706 driver31c
SOCKET_NETWORK socket1448 6707 driver31d
SOCKET_NETWORK socket1449 6708 driver31a
SOCKET_NETWORK socket1450 6709 driver31b
SOCKET_NETWORK socket1451 6710 driver31c
SOCKET_NETWORK socket1452 6711 driver31d
SOCKET_NETWORK socket1453 6712 driver31a
SOCKET_NETWORK socket1454 6713 driver31b
SOCKET_NETWORK socket1455 6714 driver31c
SOCKET_NETWORK socket1456 6715 driver31d
SOCKET_NETWORK socket1457 6716 driver31a
SOCKET_NETWORK socket1458 6717 driver31b
SOCKET_NETWORK socket1459 6718 driver31c
SOCKET_NETWORK socket1460 6719 driver31d
SOCKET_NETWORK socket1461 6720 driver31a
SOCKET_NETWORK socket1462 6721 driver31b
SOCKET_NETWORK socket1463 6722 driver31c
SOCKET_NETWORK socket1464 6723 driver31d
SOCKET_NETWORK socket1465 6724 driver31a
SOCKET_NETWORK socket1466 6725 driver31b
SOCKET_NETWORK socket1467 6726 driver31c
SOCKET_NETWORK socket1468 6727 driver31d
SOCKET_NETWORK socket1469 6728 driver31a
SOCKET_NETWORK socket1470 6729 driver31b
SOCKET_NETWORK socket1471 6730 driver31c

SOCKET_NETWORK socket1472 6731 driver31d
SOCKET_NETWORK socket1473 6732 driver31a
SOCKET_NETWORK socket1474 6733 driver31b
SOCKET_NETWORK socket1475 6734 driver31c
SOCKET_NETWORK socket1476 6735 driver31d
SOCKET_NETWORK socket1477 6736 driver31a
SOCKET_NETWORK socket1478 6737 driver31b
SOCKET_NETWORK socket1479 6738 driver31c
SOCKET_NETWORK socket1480 6739 driver31d
SOCKET_NETWORK socket1481 6740 driver31a
SOCKET_NETWORK socket1482 6741 driver31b
SOCKET_NETWORK socket1483 6742 driver31c
SOCKET_NETWORK socket1484 6743 driver31d
SOCKET_NETWORK socket1485 6744 driver31a
SOCKET_NETWORK socket1486 6745 driver31b
SOCKET_NETWORK socket1487 6746 driver31c
SOCKET_NETWORK socket1488 6747 driver31d
SOCKET_NETWORK socket1489 6748 driver32a
SOCKET_NETWORK socket1490 6749 driver32b
SOCKET_NETWORK socket1491 6750 driver32c
SOCKET_NETWORK socket1492 6751 driver32d
SOCKET_NETWORK socket1493 6752 driver32a
SOCKET_NETWORK socket1494 6753 driver32b
SOCKET_NETWORK socket1495 6754 driver32c
SOCKET_NETWORK socket1496 6755 driver32d
SOCKET_NETWORK socket1497 6756 driver32a
SOCKET_NETWORK socket1498 6757 driver32b
SOCKET_NETWORK socket1499 6758 driver32c
SOCKET_NETWORK socket1500 6759 driver32d
SOCKET_NETWORK socket1501 6760 driver32a
SOCKET_NETWORK socket1502 6761 driver32b
SOCKET_NETWORK socket1503 6762 driver32c
SOCKET_NETWORK socket1504 6763 driver32d
SOCKET_NETWORK socket1505 6764 driver32a
SOCKET_NETWORK socket1506 6765 driver32b
SOCKET_NETWORK socket1507 6766 driver32c
SOCKET_NETWORK socket1508 6767 driver32d
SOCKET_NETWORK socket1509 6768 driver32a
SOCKET_NETWORK socket1510 6769 driver32b
SOCKET_NETWORK socket1511 6770 driver32c
SOCKET_NETWORK socket1512 6771 driver32d
SOCKET_NETWORK socket1513 6772 driver32a
SOCKET_NETWORK socket1514 6773 driver32b
SOCKET_NETWORK socket1515 6774 driver32c
SOCKET_NETWORK socket1516 6775 driver32d
SOCKET_NETWORK socket1517 6776 driver32a
SOCKET_NETWORK socket1518 6777 driver32b
SOCKET_NETWORK socket1519 6778 driver32c
SOCKET_NETWORK socket1520 6779 driver32d
SOCKET_NETWORK socket1521 6780 driver32a
SOCKET_NETWORK socket1522 6781 driver32b
SOCKET_NETWORK socket1523 6782 driver32c
SOCKET_NETWORK socket1524 6783 driver32d
SOCKET_NETWORK socket1525 6784 driver32a
SOCKET_NETWORK socket1526 6785 driver32b
SOCKET_NETWORK socket1527 6786 driver32c
SOCKET_NETWORK socket1528 6787 driver32d
SOCKET_NETWORK socket1529 6788 driver32a
SOCKET_NETWORK socket1530 6789 driver32b
SOCKET_NETWORK socket1531 6790 driver32c
SOCKET_NETWORK socket1532 6791 driver32d
SOCKET_NETWORK socket1533 6792 driver32a
SOCKET_NETWORK socket1534 6793 driver32b
SOCKET_NETWORK socket1535 6794 driver32c
SOCKET_NETWORK socket1536 6795 driver32d
SOCKET_NETWORK socket1537 6700 driver33a
SOCKET_NETWORK socket1538 6701 driver33b
SOCKET_NETWORK socket1539 6702 driver33c
SOCKET_NETWORK socket1540 6703 driver33d
SOCKET_NETWORK socket1541 6704 driver33a
SOCKET_NETWORK socket1542 6705 driver33b
SOCKET_NETWORK socket1543 6706 driver33c
SOCKET_NETWORK socket1544 6707 driver33d
SOCKET_NETWORK socket1545 6708 driver33a
SOCKET_NETWORK socket1546 6709 driver33b
SOCKET_NETWORK socket1547 6710 driver33c
SOCKET_NETWORK socket1548 6711 driver33d
SOCKET_NETWORK socket1549 6712 driver33a
SOCKET_NETWORK socket1550 6713 driver33b
SOCKET_NETWORK socket1551 6714 driver33c
SOCKET_NETWORK socket1552 6715 driver33d
SOCKET_NETWORK socket1553 6716 driver33a
SOCKET_NETWORK socket1554 6717 driver33b
SOCKET_NETWORK socket1555 6718 driver33c
SOCKET_NETWORK socket1556 6719 driver33d
SOCKET_NETWORK socket1557 6720 driver33a
SOCKET_NETWORK socket1558 6721 driver33b
SOCKET_NETWORK socket1559 6722 driver33c
SOCKET_NETWORK socket1560 6723 driver33d
SOCKET_NETWORK socket1561 6724 driver33a
SOCKET_NETWORK socket1562 6725 driver33b
SOCKET_NETWORK socket1563 6726 driver33c
SOCKET_NETWORK socket1564 6727 driver33d
SOCKET_NETWORK socket1565 6728 driver33a
SOCKET_NETWORK socket1566 6729 driver33b
SOCKET_NETWORK socket1567 6730 driver33c
SOCKET_NETWORK socket1568 6731 driver33d
SOCKET_NETWORK socket1569 6732 driver33a
SOCKET_NETWORK socket1570 6733 driver33b
SOCKET_NETWORK socket1571 6734 driver33c
SOCKET_NETWORK socket1572 6735 driver33d
SOCKET_NETWORK socket1573 6736 driver33a
SOCKET_NETWORK socket1574 6737 driver33b
SOCKET_NETWORK socket1575 6738 driver33c
SOCKET_NETWORK socket1576 6739 driver33d
SOCKET_NETWORK socket1577 6740 driver33a
SOCKET_NETWORK socket1578 6741 driver33b
SOCKET_NETWORK socket1579 6742 driver33c
SOCKET_NETWORK socket1580 6743 driver33d
SOCKET_NETWORK socket1581 6744 driver33a
SOCKET_NETWORK socket1582 6745 driver33b
SOCKET_NETWORK socket1583 6746 driver33c
SOCKET_NETWORK socket1584 6747 driver33d
SOCKET_NETWORK socket1585 6748 driver34a
SOCKET_NETWORK socket1586 6749 driver34b
SOCKET_NETWORK socket1587 6750 driver34c
SOCKET_NETWORK socket1588 6751 driver34d
SOCKET_NETWORK socket1589 6752 driver34a
SOCKET_NETWORK socket1590 6753 driver34b
SOCKET_NETWORK socket1591 6754 driver34c
SOCKET_NETWORK socket1592 6755 driver34d
SOCKET_NETWORK socket1593 6756 driver34a
SOCKET_NETWORK socket1594 6757 driver34b
SOCKET_NETWORK socket1595 6758 driver34c
SOCKET_NETWORK socket1596 6759 driver34d
SOCKET_NETWORK socket1597 6760 driver34a
SOCKET_NETWORK socket1598 6761 driver34b
SOCKET_NETWORK socket1599 6762 driver34c
SOCKET_NETWORK socket1600 6763 driver34d
SOCKET_NETWORK socket1601 6764 driver34a
SOCKET_NETWORK socket1602 6765 driver34b
SOCKET_NETWORK socket1603 6766 driver34c
SOCKET_NETWORK socket1604 6767 driver34d
SOCKET_NETWORK socket1605 6768 driver34a
SOCKET_NETWORK socket1606 6769 driver34b
SOCKET_NETWORK socket1607 6770 driver34c
SOCKET_NETWORK socket1608 6771 driver34d
SOCKET_NETWORK socket1609 6772 driver34a
SOCKET_NETWORK socket1610 6773 driver34b
SOCKET_NETWORK socket1611 6774 driver34c
SOCKET_NETWORK socket1612 6775 driver34d
SOCKET_NETWORK socket1613 6776 driver34a
SOCKET_NETWORK socket1614 6777 driver34b
SOCKET_NETWORK socket1615 6778 driver34c
SOCKET_NETWORK socket1616 6779 driver34d
SOCKET_NETWORK socket1617 6780 driver34a
SOCKET_NETWORK socket1618 6781 driver34b
SOCKET_NETWORK socket1619 6782 driver34c
SOCKET_NETWORK socket1620 6783 driver34d
SOCKET_NETWORK socket1621 6784 driver34a
SOCKET_NETWORK socket1622 6785 driver34b
SOCKET_NETWORK socket1623 6786 driver34c
SOCKET_NETWORK socket1624 6787 driver34d
SOCKET_NETWORK socket1625 6788 driver34a
SOCKET_NETWORK socket1626 6789 driver34b
SOCKET_NETWORK socket1627 6790 driver34c
SOCKET_NETWORK socket1628 6791 driver34d
SOCKET_NETWORK socket1629 6792 driver34a
SOCKET_NETWORK socket1630 6793 driver34b
SOCKET_NETWORK socket1631 6794 driver34c
SOCKET_NETWORK socket1632 6795 driver34d
SOCKET_NETWORK socket1633 6700 driver35a
SOCKET_NETWORK socket1634 6701 driver35b
SOCKET_NETWORK socket1635 6702 driver35c
SOCKET_NETWORK socket1636 6703 driver35d
SOCKET_NETWORK socket1637 6704 driver35a
SOCKET_NETWORK socket1638 6705 driver35b
SOCKET_NETWORK socket1639 6706 driver35c
SOCKET_NETWORK socket1640 6707 driver35d
SOCKET_NETWORK socket1641 6708 driver35a
SOCKET_NETWORK socket1642 6709 driver35b
SOCKET_NETWORK socket1643 6710 driver35c
SOCKET_NETWORK socket1644 6711 driver35d
SOCKET_NETWORK socket1645 6712 driver35a
SOCKET_NETWORK socket1646 6713 driver35b
SOCKET_NETWORK socket1647 6714 driver35c
SOCKET_NETWORK socket1648 6715 driver35d
SOCKET_NETWORK socket1649 6716 driver35a
SOCKET_NETWORK socket1650 6717 driver35b
SOCKET_NETWORK socket1651 6718 driver35c
SOCKET_NETWORK socket1652 6719 driver35d
SOCKET_NETWORK socket1653 6720 driver35a
SOCKET_NETWORK socket1654 6721 driver35b
SOCKET_NETWORK socket1655 6722 driver35c
SOCKET_NETWORK socket1656 6723 driver35d
SOCKET_NETWORK socket1657 6724 driver35a
SOCKET_NETWORK socket1658 6725 driver35b
SOCKET_NETWORK socket1659 6726 driver35c
SOCKET_NETWORK socket1660 6727 driver35d
SOCKET_NETWORK socket1661 6728 driver35a
SOCKET_NETWORK socket1662 6729 driver35b
SOCKET_NETWORK socket1663 6730 driver35c
SOCKET_NETWORK socket1664 6731 driver35d
SOCKET_NETWORK socket1665 6732 driver35a
SOCKET_NETWORK socket1666 6733 driver35b
SOCKET_NETWORK socket1667 6734 driver35c
SOCKET_NETWORK socket1668 6735 driver35d
SOCKET_NETWORK socket1669 6736 driver35a

```
SOCKET_NETWORK socket1670  6737 driver35b
SOCKET_NETWORK socket1671  6738 driver35c
SOCKET_NETWORK socket1672  6739 driver35d
SOCKET_NETWORK socket1673  6740 driver35a
SOCKET_NETWORK socket1674  6741 driver35b
SOCKET_NETWORK socket1675  6742 driver35c
SOCKET_NETWORK socket1676  6743 driver35d
SOCKET_NETWORK socket1677  6744 driver35a
SOCKET_NETWORK socket1678  6745 driver35b
SOCKET_NETWORK socket1679  6746 driver35c
SOCKET_NETWORK socket1680  6747 driver35d
SOCKET_NETWORK socket1681  6748 driver36a
SOCKET_NETWORK socket1682  6749 driver36b
SOCKET_NETWORK socket1683  6750 driver36c
SOCKET_NETWORK socket1684  6751 driver36d
SOCKET_NETWORK socket1685  6752 driver36a
SOCKET_NETWORK socket1686  6753 driver36b
SOCKET_NETWORK socket1687  6754 driver36c
SOCKET_NETWORK socket1688  6755 driver36d
SOCKET_NETWORK socket1689  6756 driver36a
SOCKET_NETWORK socket1690  6757 driver36b
SOCKET_NETWORK socket1691  6758 driver36c
SOCKET_NETWORK socket1692  6759 driver36d
SOCKET_NETWORK socket1693  6760 driver36a
SOCKET_NETWORK socket1694  6761 driver36b
SOCKET_NETWORK socket1695  6762 driver36c
SOCKET_NETWORK socket1696  6763 driver36d
SOCKET_NETWORK socket1697  6764 driver36a
SOCKET_NETWORK socket1698  6765 driver36b
SOCKET_NETWORK socket1699  6766 driver36c
SOCKET_NETWORK socket1700  6767 driver36d
SOCKET_NETWORK socket1701  6768 driver36a
SOCKET_NETWORK socket1702  6769 driver36b
SOCKET_NETWORK socket1703  6770 driver36c
SOCKET_NETWORK socket1704  6771 driver36d
SOCKET_NETWORK socket1705  6772 driver36a
SOCKET_NETWORK socket1706  6773 driver36b
SOCKET_NETWORK socket1707  6774 driver36c
SOCKET_NETWORK socket1708  6775 driver36d
SOCKET_NETWORK socket1709  6776 driver36a
SOCKET_NETWORK socket1710  6777 driver36b
SOCKET_NETWORK socket1711  6778 driver36c
SOCKET_NETWORK socket1712  6779 driver36d
SOCKET_NETWORK socket1713  6780 driver36a
SOCKET_NETWORK socket1714  6781 driver36b
SOCKET_NETWORK socket1715  6782 driver36c
SOCKET_NETWORK socket1716  6783 driver36d
SOCKET_NETWORK socket1717  6784 driver36a
SOCKET_NETWORK socket1718  6785 driver36b
SOCKET_NETWORK socket1719  6786 driver36c
SOCKET_NETWORK socket1720  6787 driver36d
SOCKET_NETWORK socket1721  6788 driver36a
SOCKET_NETWORK socket1722  6789 driver36b
SOCKET_NETWORK socket1723  6790 driver36c
SOCKET_NETWORK socket1724  6791 driver36d
SOCKET_NETWORK socket1725  6792 driver36a
SOCKET_NETWORK socket1726  6793 driver36b
SOCKET_NETWORK socket1727  6794 driver36c
SOCKET_NETWORK socket1728  6795 driver36d
#elif MASTER_NUM7
SOCKET_NETWORK socket1729  6700 driver37a
SOCKET_NETWORK socket1730  6701 driver37b
SOCKET_NETWORK socket1731  6702 driver37c
SOCKET_NETWORK socket1732  6703 driver37d
SOCKET_NETWORK socket1733  6704 driver37a
SOCKET_NETWORK socket1734  6705 driver37b
SOCKET_NETWORK socket1735  6706 driver37c
SOCKET_NETWORK socket1736  6707 driver37d
SOCKET_NETWORK socket1737  6708 driver37a
SOCKET_NETWORK socket1738  6709 driver37b
SOCKET_NETWORK socket1739  6710 driver37c
SOCKET_NETWORK socket1740  6711 driver37d
SOCKET_NETWORK socket1741  6712 driver37a
SOCKET_NETWORK socket1742  6713 driver37b
SOCKET_NETWORK socket1743  6714 driver37c
SOCKET_NETWORK socket1744  6715 driver37d
SOCKET_NETWORK socket1745  6716 driver37a
SOCKET_NETWORK socket1746  6717 driver37b
SOCKET_NETWORK socket1747  6718 driver37c
SOCKET_NETWORK socket1748  6719 driver37d
SOCKET_NETWORK socket1749  6720 driver37a
SOCKET_NETWORK socket1750  6721 driver37b
SOCKET_NETWORK socket1751  6722 driver37c
SOCKET_NETWORK socket1752  6723 driver37d
SOCKET_NETWORK socket1753  6724 driver37a
SOCKET_NETWORK socket1754  6725 driver37b
SOCKET_NETWORK socket1755  6726 driver37c
SOCKET_NETWORK socket1756  6727 driver37d
SOCKET_NETWORK socket1757  6728 driver37a
SOCKET_NETWORK socket1758  6729 driver37b
SOCKET_NETWORK socket1759  6730 driver37c
SOCKET_NETWORK socket1760  6731 driver37d
SOCKET_NETWORK socket1761  6732 driver37a
SOCKET_NETWORK socket1762  6733 driver37b
SOCKET_NETWORK socket1763  6734 driver37c
SOCKET_NETWORK socket1764  6735 driver37d
SOCKET_NETWORK socket1765  6736 driver37a
SOCKET_NETWORK socket1766  6737 driver37b
SOCKET_NETWORK socket1767  6738 driver37c
SOCKET_NETWORK socket1768  6739 driver37d
SOCKET_NETWORK socket1769  6740 driver37a
SOCKET_NETWORK socket1770  6741 driver37b
SOCKET_NETWORK socket1771  6742 driver37c
SOCKET_NETWORK socket1772  6743 driver37d
SOCKET_NETWORK socket1773  6744 driver37a
SOCKET_NETWORK socket1774  6745 driver37b
SOCKET_NETWORK socket1775  6746 driver37c
SOCKET_NETWORK socket1776  6747 driver37d
SOCKET_NETWORK socket1777  6748 driver38a
SOCKET_NETWORK socket1778  6749 driver38b
SOCKET_NETWORK socket1779  6750 driver38c
SOCKET_NETWORK socket1780  6751 driver38d
SOCKET_NETWORK socket1781  6752 driver38a
SOCKET_NETWORK socket1782  6753 driver38b
SOCKET_NETWORK socket1783  6754 driver38c
SOCKET_NETWORK socket1784  6755 driver38d
SOCKET_NETWORK socket1785  6756 driver38a
SOCKET_NETWORK socket1786  6757 driver38b
SOCKET_NETWORK socket1787  6758 driver38c
SOCKET_NETWORK socket1788  6759 driver38d
SOCKET_NETWORK socket1789  6760 driver38a
SOCKET_NETWORK socket1790  6761 driver38b
SOCKET_NETWORK socket1791  6762 driver38c
SOCKET_NETWORK socket1792  6763 driver38d
SOCKET_NETWORK socket1793  6764 driver38a
SOCKET_NETWORK socket1794  6765 driver38b
SOCKET_NETWORK socket1795  6766 driver38c
SOCKET_NETWORK socket1796  6767 driver38d
SOCKET_NETWORK socket1797  6768 driver38a
SOCKET_NETWORK socket1798  6769 driver38b
SOCKET_NETWORK socket1799  6770 driver38c
SOCKET_NETWORK socket1800  6771 driver38d
SOCKET_NETWORK socket1801  6772 driver38a
SOCKET_NETWORK socket1802  6773 driver38b
SOCKET_NETWORK socket1803  6774 driver38c
SOCKET_NETWORK socket1804  6775 driver38d
SOCKET_NETWORK socket1805  6776 driver38a
SOCKET_NETWORK socket1806  6777 driver38b
SOCKET_NETWORK socket1807  6778 driver38c
SOCKET_NETWORK socket1808  6779 driver38d
SOCKET_NETWORK socket1809  6780 driver38a
SOCKET_NETWORK socket1810  6781 driver38b
SOCKET_NETWORK socket1811  6782 driver38c
SOCKET_NETWORK socket1812  6783 driver38d
SOCKET_NETWORK socket1813  6784 driver38a
SOCKET_NETWORK socket1814  6785 driver38b
SOCKET_NETWORK socket1815  6786 driver38c
SOCKET_NETWORK socket1816  6787 driver38d
SOCKET_NETWORK socket1817  6788 driver38a
SOCKET_NETWORK socket1818  6789 driver38b
SOCKET_NETWORK socket1819  6790 driver38c
SOCKET_NETWORK socket1820  6791 driver38d
SOCKET_NETWORK socket1821  6792 driver38a
SOCKET_NETWORK socket1822  6793 driver38b
SOCKET_NETWORK socket1823  6794 driver38c
SOCKET_NETWORK socket1824  6795 driver38d
SOCKET_NETWORK socket1825  6700 driver39a
SOCKET_NETWORK socket1826  6701 driver39b
SOCKET_NETWORK socket1827  6702 driver39c
SOCKET_NETWORK socket1828  6703 driver39d
SOCKET_NETWORK socket1829  6704 driver39a
SOCKET_NETWORK socket1830  6705 driver39b
SOCKET_NETWORK socket1831  6706 driver39c
SOCKET_NETWORK socket1832  6707 driver39d
SOCKET_NETWORK socket1833  6708 driver39a
SOCKET_NETWORK socket1834  6709 driver39b
SOCKET_NETWORK socket1835  6710 driver39c
SOCKET_NETWORK socket1836  6711 driver39d
SOCKET_NETWORK socket1837  6712 driver39a
SOCKET_NETWORK socket1838  6713 driver39b
SOCKET_NETWORK socket1839  6714 driver39c
SOCKET_NETWORK socket1840  6715 driver39d
SOCKET_NETWORK socket1841  6716 driver39a
SOCKET_NETWORK socket1842  6717 driver39b
SOCKET_NETWORK socket1843  6718 driver39c
SOCKET_NETWORK socket1844  6719 driver39d
SOCKET_NETWORK socket1845  6720 driver39a
SOCKET_NETWORK socket1846  6721 driver39b
SOCKET_NETWORK socket1847  6722 driver39c
SOCKET_NETWORK socket1848  6723 driver39d
SOCKET_NETWORK socket1849  6724 driver39a
SOCKET_NETWORK socket1850  6725 driver39b
SOCKET_NETWORK socket1851  6726 driver39c
SOCKET_NETWORK socket1852  6727 driver39d
SOCKET_NETWORK socket1853  6728 driver39a
SOCKET_NETWORK socket1854  6729 driver39b
SOCKET_NETWORK socket1855  6730 driver39c
SOCKET_NETWORK socket1856  6731 driver39d
SOCKET_NETWORK socket1857  6732 driver39a
SOCKET_NETWORK socket1858  6733 driver39b
SOCKET_NETWORK socket1859  6734 driver39c
SOCKET_NETWORK socket1860  6735 driver39d
SOCKET_NETWORK socket1861  6736 driver39a
SOCKET_NETWORK socket1862  6737 driver39b
SOCKET_NETWORK socket1863  6738 driver39c
SOCKET_NETWORK socket1864  6739 driver39d
SOCKET_NETWORK socket1865  6740 driver39a
SOCKET_NETWORK socket1866  6741 driver39b
```

SOCKET_NETWORK socket1867 6742 driver39c
SOCKET_NETWORK socket1868 6743 driver39d
SOCKET_NETWORK socket1869 6744 driver39a
SOCKET_NETWORK socket1870 6745 driver39b
SOCKET_NETWORK socket1871 6746 driver39c
SOCKET_NETWORK socket1872 6747 driver39d
SOCKET_NETWORK socket1873 6748 driver40a
SOCKET_NETWORK socket1874 6749 driver40b
SOCKET_NETWORK socket1875 6750 driver40c
SOCKET_NETWORK socket1876 6751 driver40d
SOCKET_NETWORK socket1877 6752 driver40a
SOCKET_NETWORK socket1878 6753 driver40b
SOCKET_NETWORK socket1879 6754 driver40c
SOCKET_NETWORK socket1880 6755 driver40d
SOCKET_NETWORK socket1881 6756 driver40a
SOCKET_NETWORK socket1882 6757 driver40b
SOCKET_NETWORK socket1883 6758 driver40c
SOCKET_NETWORK socket1884 6759 driver40d
SOCKET_NETWORK socket1885 6760 driver40a
SOCKET_NETWORK socket1886 6761 driver40b
SOCKET_NETWORK socket1887 6762 driver40c
SOCKET_NETWORK socket1888 6763 driver40d
SOCKET_NETWORK socket1889 6764 driver40a
SOCKET_NETWORK socket1890 6765 driver40b
SOCKET_NETWORK socket1891 6766 driver40c
SOCKET_NETWORK socket1892 6767 driver40d
SOCKET_NETWORK socket1893 6768 driver40a
SOCKET_NETWORK socket1894 6769 driver40b
SOCKET_NETWORK socket1895 6770 driver40c
SOCKET_NETWORK socket1896 6771 driver40d
SOCKET_NETWORK socket1897 6772 driver40a
SOCKET_NETWORK socket1898 6773 driver40b
SOCKET_NETWORK socket1899 6774 driver40c
SOCKET_NETWORK socket1900 6775 driver40d
SOCKET_NETWORK socket1901 6776 driver40a
SOCKET_NETWORK socket1902 6777 driver40b
SOCKET_NETWORK socket1903 6778 driver40c
SOCKET_NETWORK socket1904 6779 driver40d
SOCKET_NETWORK socket1905 6780 driver40a
SOCKET_NETWORK socket1906 6781 driver40b
SOCKET_NETWORK socket1907 6782 driver40c
SOCKET_NETWORK socket1908 6783 driver40d
SOCKET_NETWORK socket1909 6784 driver40a
SOCKET_NETWORK socket1910 6785 driver40b
SOCKET_NETWORK socket1911 6786 driver40c
SOCKET_NETWORK socket1912 6787 driver40d
SOCKET_NETWORK socket1913 6788 driver40a
SOCKET_NETWORK socket1914 6789 driver40b
SOCKET_NETWORK socket1915 6790 driver40c
SOCKET_NETWORK socket1916 6791 driver40d
SOCKET_NETWORK socket1917 6792 driver40a
SOCKET_NETWORK socket1918 6793 driver40b
SOCKET_NETWORK socket1919 6794 driver40c
SOCKET_NETWORK socket1920 6795 driver40d
SOCKET_NETWORK socket1921 6700 driver41a
SOCKET_NETWORK socket1922 6701 driver41b
SOCKET_NETWORK socket1923 6702 driver41c
SOCKET_NETWORK socket1924 6703 driver41d
SOCKET_NETWORK socket1925 6704 driver41a
SOCKET_NETWORK socket1926 6705 driver41b
SOCKET_NETWORK socket1927 6706 driver41c
SOCKET_NETWORK socket1928 6707 driver41d
SOCKET_NETWORK socket1929 6708 driver41a
SOCKET_NETWORK socket1930 6709 driver41b
SOCKET_NETWORK socket1931 6710 driver41c
SOCKET_NETWORK socket1932 6711 driver41d
SOCKET_NETWORK socket1933 6712 driver41a
SOCKET_NETWORK socket1934 6713 driver41b
SOCKET_NETWORK socket1935 6714 driver41c
SOCKET_NETWORK socket1936 6715 driver41d
SOCKET_NETWORK socket1937 6716 driver41a
SOCKET_NETWORK socket1938 6717 driver41b
SOCKET_NETWORK socket1939 6718 driver41c
SOCKET_NETWORK socket1940 6719 driver41d
SOCKET_NETWORK socket1941 6720 driver41a
SOCKET_NETWORK socket1942 6721 driver41b
SOCKET_NETWORK socket1943 6722 driver41c
SOCKET_NETWORK socket1944 6723 driver41d
SOCKET_NETWORK socket1945 6724 driver41a
SOCKET_NETWORK socket1946 6725 driver41b
SOCKET_NETWORK socket1947 6726 driver41c
SOCKET_NETWORK socket1948 6727 driver41d
SOCKET_NETWORK socket1949 6728 driver41a
SOCKET_NETWORK socket1950 6729 driver41b
SOCKET_NETWORK socket1951 6730 driver41c
SOCKET_NETWORK socket1952 6731 driver41d
SOCKET_NETWORK socket1953 6732 driver41a
SOCKET_NETWORK socket1954 6733 driver41b
SOCKET_NETWORK socket1955 6734 driver41c
SOCKET_NETWORK socket1956 6735 driver41d
SOCKET_NETWORK socket1957 6736 driver41a
SOCKET_NETWORK socket1958 6737 driver41b
SOCKET_NETWORK socket1959 6738 driver41c
SOCKET_NETWORK socket1960 6739 driver41d
SOCKET_NETWORK socket1961 6740 driver41a
SOCKET_NETWORK socket1962 6741 driver41b
SOCKET_NETWORK socket1963 6742 driver41c
SOCKET_NETWORK socket1964 6743 driver41d
SOCKET_NETWORK socket1965 6744 driver41a
SOCKET_NETWORK socket1966 6745 driver41b
SOCKET_NETWORK socket1967 6746 driver41c
SOCKET_NETWORK socket1968 6747 driver41d
SOCKET_NETWORK socket1969 6748 driver42a
SOCKET_NETWORK socket1970 6749 driver42b
SOCKET_NETWORK socket1971 6750 driver42c
SOCKET_NETWORK socket1972 6751 driver42d
SOCKET_NETWORK socket1973 6752 driver42a
SOCKET_NETWORK socket1974 6753 driver42b
SOCKET_NETWORK socket1975 6754 driver42c
SOCKET_NETWORK socket1976 6755 driver42d
SOCKET_NETWORK socket1977 6756 driver42a
SOCKET_NETWORK socket1978 6757 driver42b
SOCKET_NETWORK socket1979 6758 driver42c
SOCKET_NETWORK socket1980 6759 driver42d
SOCKET_NETWORK socket1981 6760 driver42a
SOCKET_NETWORK socket1982 6761 driver42b
SOCKET_NETWORK socket1983 6762 driver42c
SOCKET_NETWORK socket1984 6763 driver42d
SOCKET_NETWORK socket1985 6764 driver42a
SOCKET_NETWORK socket1986 6765 driver42b
SOCKET_NETWORK socket1987 6766 driver42c
SOCKET_NETWORK socket1988 6767 driver42d
SOCKET_NETWORK socket1989 6768 driver42a
SOCKET_NETWORK socket1990 6769 driver42b
SOCKET_NETWORK socket1991 6770 driver42c
SOCKET_NETWORK socket1992 6771 driver42d
SOCKET_NETWORK socket1993 6772 driver42a
SOCKET_NETWORK socket1994 6773 driver42b
SOCKET_NETWORK socket1995 6774 driver42c
SOCKET_NETWORK socket1996 6775 driver42d
SOCKET_NETWORK socket1997 6776 driver42a
SOCKET_NETWORK socket1998 6777 driver42b
SOCKET_NETWORK socket1999 6778 driver42c
SOCKET_NETWORK socket2000 6779 driver42d
SOCKET_NETWORK socket2001 6780 driver42a
SOCKET_NETWORK socket2002 6781 driver42b
SOCKET_NETWORK socket2003 6782 driver42c
SOCKET_NETWORK socket2004 6783 driver42d
SOCKET_NETWORK socket2005 6784 driver42a
SOCKET_NETWORK socket2006 6785 driver42b
SOCKET_NETWORK socket2007 6786 driver42c
SOCKET_NETWORK socket2008 6787 driver42d
SOCKET_NETWORK socket2009 6788 driver42a
SOCKET_NETWORK socket2010 6789 driver42b
SOCKET_NETWORK socket2011 6790 driver42c
SOCKET_NETWORK socket2012 6791 driver42d
SOCKET_NETWORK socket2013 6792 driver42a
SOCKET_NETWORK socket2014 6793 driver42b
SOCKET_NETWORK socket2015 6794 driver42c
SOCKET_NETWORK socket2016 6795 driver42d
#elif MASTER_NUM8
SOCKET_NETWORK socket2017 6700 driver43a
SOCKET_NETWORK socket2018 6701 driver43b
SOCKET_NETWORK socket2019 6702 driver43c
SOCKET_NETWORK socket2020 6703 driver43d
SOCKET_NETWORK socket2021 6704 driver43a
SOCKET_NETWORK socket2022 6705 driver43b
SOCKET_NETWORK socket2023 6706 driver43c
SOCKET_NETWORK socket2024 6707 driver43d
SOCKET_NETWORK socket2025 6708 driver43a
SOCKET_NETWORK socket2026 6709 driver43b
SOCKET_NETWORK socket2027 6710 driver43c
SOCKET_NETWORK socket2028 6711 driver43d
SOCKET_NETWORK socket2029 6712 driver43a
SOCKET_NETWORK socket2030 6713 driver43b
SOCKET_NETWORK socket2031 6714 driver43c
SOCKET_NETWORK socket2032 6715 driver43d
SOCKET_NETWORK socket2033 6716 driver43a
SOCKET_NETWORK socket2034 6717 driver43b
SOCKET_NETWORK socket2035 6718 driver43c
SOCKET_NETWORK socket2036 6719 driver43d
SOCKET_NETWORK socket2037 6720 driver43a
SOCKET_NETWORK socket2038 6721 driver43b
SOCKET_NETWORK socket2039 6722 driver43c
SOCKET_NETWORK socket2040 6723 driver43d
SOCKET_NETWORK socket2041 6724 driver43a
SOCKET_NETWORK socket2042 6725 driver43b
SOCKET_NETWORK socket2043 6726 driver43c
SOCKET_NETWORK socket2044 6727 driver43d
SOCKET_NETWORK socket2045 6728 driver43a
SOCKET_NETWORK socket2046 6729 driver43b
SOCKET_NETWORK socket2047 6730 driver43c
SOCKET_NETWORK socket2048 6731 driver43d
SOCKET_NETWORK socket2049 6732 driver43a
SOCKET_NETWORK socket2050 6733 driver43b
SOCKET_NETWORK socket2051 6734 driver43c
SOCKET_NETWORK socket2052 6735 driver43d
SOCKET_NETWORK socket2053 6736 driver43a
SOCKET_NETWORK socket2054 6737 driver43b
SOCKET_NETWORK socket2055 6738 driver43c
SOCKET_NETWORK socket2056 6739 driver43d
SOCKET_NETWORK socket2057 6740 driver43a
SOCKET_NETWORK socket2058 6741 driver43b
SOCKET_NETWORK socket2059 6742 driver43c
SOCKET_NETWORK socket2060 6743 driver43d
SOCKET_NETWORK socket2061 6744 driver43a
SOCKET_NETWORK socket2062 6745 driver43b
SOCKET_NETWORK socket2063 6746 driver43c

SOCKET_NETWORK socket2064 6747 driver43d
SOCKET_NETWORK socket2065 6748 driver44a
SOCKET_NETWORK socket2066 6749 driver44b
SOCKET_NETWORK socket2067 6750 driver44c
SOCKET_NETWORK socket2068 6751 driver44d
SOCKET_NETWORK socket2069 6752 driver44a
SOCKET_NETWORK socket2070 6753 driver44b
SOCKET_NETWORK socket2071 6754 driver44c
SOCKET_NETWORK socket2072 6755 driver44d
SOCKET_NETWORK socket2073 6756 driver44a
SOCKET_NETWORK socket2074 6757 driver44b
SOCKET_NETWORK socket2075 6758 driver44c
SOCKET_NETWORK socket2076 6759 driver44d
SOCKET_NETWORK socket2077 6760 driver44a
SOCKET_NETWORK socket2078 6761 driver44b
SOCKET_NETWORK socket2079 6762 driver44c
SOCKET_NETWORK socket2080 6763 driver44d
SOCKET_NETWORK socket2081 6764 driver44a
SOCKET_NETWORK socket2082 6765 driver44b
SOCKET_NETWORK socket2083 6766 driver44c
SOCKET_NETWORK socket2084 6767 driver44d
SOCKET_NETWORK socket2085 6768 driver44a
SOCKET_NETWORK socket2086 6769 driver44b
SOCKET_NETWORK socket2087 6770 driver44c
SOCKET_NETWORK socket2088 6771 driver44d
SOCKET_NETWORK socket2089 6772 driver44a
SOCKET_NETWORK socket2090 6773 driver44b
SOCKET_NETWORK socket2091 6774 driver44c
SOCKET_NETWORK socket2092 6775 driver44d
SOCKET_NETWORK socket2093 6776 driver44a
SOCKET_NETWORK socket2094 6777 driver44b
SOCKET_NETWORK socket2095 6778 driver44c
SOCKET_NETWORK socket2096 6779 driver44d
SOCKET_NETWORK socket2097 6780 driver44a
SOCKET_NETWORK socket2098 6781 driver44b
SOCKET_NETWORK socket2099 6782 driver44c
SOCKET_NETWORK socket2100 6783 driver44d
SOCKET_NETWORK socket2101 6784 driver44a
SOCKET_NETWORK socket2102 6785 driver44b
SOCKET_NETWORK socket2103 6786 driver44c
SOCKET_NETWORK socket2104 6787 driver44d
SOCKET_NETWORK socket2105 6788 driver44a
SOCKET_NETWORK socket2106 6789 driver44b
SOCKET_NETWORK socket2107 6790 driver44c
SOCKET_NETWORK socket2108 6791 driver44d
SOCKET_NETWORK socket2109 6792 driver44a
SOCKET_NETWORK socket2110 6793 driver44b
SOCKET_NETWORK socket2111 6794 driver44c
SOCKET_NETWORK socket2112 6795 driver44d
SOCKET_NETWORK socket2113 6700 driver45a
SOCKET_NETWORK socket2114 6701 driver45b
SOCKET_NETWORK socket2115 6702 driver45c
SOCKET_NETWORK socket2116 6703 driver45d
SOCKET_NETWORK socket2117 6704 driver45a
SOCKET_NETWORK socket2118 6705 driver45b
SOCKET_NETWORK socket2119 6706 driver45c
SOCKET_NETWORK socket2120 6707 driver45d
SOCKET_NETWORK socket2121 6708 driver45a
SOCKET_NETWORK socket2122 6709 driver45b
SOCKET_NETWORK socket2123 6710 driver45c
SOCKET_NETWORK socket2124 6711 driver45d
SOCKET_NETWORK socket2125 6712 driver45a
SOCKET_NETWORK socket2126 6713 driver45b
SOCKET_NETWORK socket2127 6714 driver45c
SOCKET_NETWORK socket2128 6715 driver45d
SOCKET_NETWORK socket2129 6716 driver45a
SOCKET_NETWORK socket2130 6717 driver45b
SOCKET_NETWORK socket2131 6718 driver45c
SOCKET_NETWORK socket2132 6719 driver45d
SOCKET_NETWORK socket2133 6720 driver45a
SOCKET_NETWORK socket2134 6721 driver45b
SOCKET_NETWORK socket2135 6722 driver45c
SOCKET_NETWORK socket2136 6723 driver45d
SOCKET_NETWORK socket2137 6724 driver45a
SOCKET_NETWORK socket2138 6725 driver45b
SOCKET_NETWORK socket2139 6726 driver45c
SOCKET_NETWORK socket2140 6727 driver45d
SOCKET_NETWORK socket2141 6728 driver45a
SOCKET_NETWORK socket2142 6729 driver45b
SOCKET_NETWORK socket2143 6730 driver45c
SOCKET_NETWORK socket2144 6731 driver45d
SOCKET_NETWORK socket2145 6732 driver45a
SOCKET_NETWORK socket2146 6733 driver45b
SOCKET_NETWORK socket2147 6734 driver45c
SOCKET_NETWORK socket2148 6735 driver45d
SOCKET_NETWORK socket2149 6736 driver45a
SOCKET_NETWORK socket2150 6737 driver45b
SOCKET_NETWORK socket2151 6738 driver45c
SOCKET_NETWORK socket2152 6739 driver45d
SOCKET_NETWORK socket2153 6740 driver45a
SOCKET_NETWORK socket2154 6741 driver45b
SOCKET_NETWORK socket2155 6742 driver45c
SOCKET_NETWORK socket2156 6743 driver45d
SOCKET_NETWORK socket2157 6744 driver45a
SOCKET_NETWORK socket2158 6745 driver45b
SOCKET_NETWORK socket2159 6746 driver45c
SOCKET_NETWORK socket2160 6747 driver45d
SOCKET_NETWORK socket2161 6748 driver46a
SOCKET_NETWORK socket2162 6749 driver46b
SOCKET_NETWORK socket2163 6750 driver46c
SOCKET_NETWORK socket2164 6751 driver46d
SOCKET_NETWORK socket2165 6752 driver46a
SOCKET_NETWORK socket2166 6753 driver46b
SOCKET_NETWORK socket2167 6754 driver46c
SOCKET_NETWORK socket2168 6755 driver46d
SOCKET_NETWORK socket2169 6756 driver46a
SOCKET_NETWORK socket2170 6757 driver46b
SOCKET_NETWORK socket2171 6758 driver46c
SOCKET_NETWORK socket2172 6759 driver46d
SOCKET_NETWORK socket2173 6760 driver46a
SOCKET_NETWORK socket2174 6761 driver46b
SOCKET_NETWORK socket2175 6762 driver46c
SOCKET_NETWORK socket2176 6763 driver46d
SOCKET_NETWORK socket2177 6764 driver46a
SOCKET_NETWORK socket2178 6765 driver46b
SOCKET_NETWORK socket2179 6766 driver46c
SOCKET_NETWORK socket2180 6767 driver46d
SOCKET_NETWORK socket2181 6768 driver46a
SOCKET_NETWORK socket2182 6769 driver46b
SOCKET_NETWORK socket2183 6770 driver46c
SOCKET_NETWORK socket2184 6771 driver46d
SOCKET_NETWORK socket2185 6772 driver46a
SOCKET_NETWORK socket2186 6773 driver46b
SOCKET_NETWORK socket2187 6774 driver46c
SOCKET_NETWORK socket2188 6775 driver46d
SOCKET_NETWORK socket2189 6776 driver46a
SOCKET_NETWORK socket2190 6777 driver46b
SOCKET_NETWORK socket2191 6778 driver46c
SOCKET_NETWORK socket2192 6779 driver46d
SOCKET_NETWORK socket2193 6780 driver46a
SOCKET_NETWORK socket2194 6781 driver46b
SOCKET_NETWORK socket2195 6782 driver46c
SOCKET_NETWORK socket2196 6783 driver46d
SOCKET_NETWORK socket2197 6784 driver46a
SOCKET_NETWORK socket2198 6785 driver46b
SOCKET_NETWORK socket2199 6786 driver46c
SOCKET_NETWORK socket2200 6787 driver46d
SOCKET_NETWORK socket2201 6788 driver46a
SOCKET_NETWORK socket2202 6789 driver46b
SOCKET_NETWORK socket2203 6790 driver46c
SOCKET_NETWORK socket2204 6791 driver46d
SOCKET_NETWORK socket2205 6792 driver46a
SOCKET_NETWORK socket2206 6793 driver46b
SOCKET_NETWORK socket2207 6794 driver46c
SOCKET_NETWORK socket2208 6795 driver46d
SOCKET_NETWORK socket2209 6700 driver47a
SOCKET_NETWORK socket2210 6701 driver47b
SOCKET_NETWORK socket2211 6702 driver47c
SOCKET_NETWORK socket2212 6703 driver47d
SOCKET_NETWORK socket2213 6704 driver47a
SOCKET_NETWORK socket2214 6705 driver47b
SOCKET_NETWORK socket2215 6706 driver47c
SOCKET_NETWORK socket2216 6707 driver47d
SOCKET_NETWORK socket2217 6708 driver47a
SOCKET_NETWORK socket2218 6709 driver47b
SOCKET_NETWORK socket2219 6710 driver47c
SOCKET_NETWORK socket2220 6711 driver47d
SOCKET_NETWORK socket2221 6712 driver47a
SOCKET_NETWORK socket2222 6713 driver47b
SOCKET_NETWORK socket2223 6714 driver47c
SOCKET_NETWORK socket2224 6715 driver47d
SOCKET_NETWORK socket2225 6716 driver47a
SOCKET_NETWORK socket2226 6717 driver47b
SOCKET_NETWORK socket2227 6718 driver47c
SOCKET_NETWORK socket2228 6719 driver47d
SOCKET_NETWORK socket2229 6720 driver47a
SOCKET_NETWORK socket2230 6721 driver47b
SOCKET_NETWORK socket2231 6722 driver47c
SOCKET_NETWORK socket2232 6723 driver47d
SOCKET_NETWORK socket2233 6724 driver47a
SOCKET_NETWORK socket2234 6725 driver47b
SOCKET_NETWORK socket2235 6726 driver47c
SOCKET_NETWORK socket2236 6727 driver47d
SOCKET_NETWORK socket2237 6728 driver47a
SOCKET_NETWORK socket2238 6729 driver47b
SOCKET_NETWORK socket2239 6730 driver47c
SOCKET_NETWORK socket2240 6731 driver47d
SOCKET_NETWORK socket2241 6732 driver47a
SOCKET_NETWORK socket2242 6733 driver47b
SOCKET_NETWORK socket2243 6734 driver47c
SOCKET_NETWORK socket2244 6735 driver47d
SOCKET_NETWORK socket2245 6736 driver47a
SOCKET_NETWORK socket2246 6737 driver47b
SOCKET_NETWORK socket2247 6738 driver47c
SOCKET_NETWORK socket2248 6739 driver47d
SOCKET_NETWORK socket2249 6740 driver47a
SOCKET_NETWORK socket2250 6741 driver47b
SOCKET_NETWORK socket2251 6742 driver47c
SOCKET_NETWORK socket2252 6743 driver47d
SOCKET_NETWORK socket2253 6744 driver47a
SOCKET_NETWORK socket2254 6745 driver47b
SOCKET_NETWORK socket2255 6746 driver47c
SOCKET_NETWORK socket2256 6747 driver47d
SOCKET_NETWORK socket2257 6748 driver48a
SOCKET_NETWORK socket2258 6749 driver48b
SOCKET_NETWORK socket2259 6750 driver48c
SOCKET_NETWORK socket2260 6751 driver48d
SOCKET_NETWORK socket2261 6752 driver48a

SOCKET_NETWORK socket2262 6753 driver48b
SOCKET_NETWORK socket2263 6754 driver48c
SOCKET_NETWORK socket2264 6755 driver48d
SOCKET_NETWORK socket2265 6756 driver48a
SOCKET_NETWORK socket2266 6757 driver48b
SOCKET_NETWORK socket2267 6758 driver48c
SOCKET_NETWORK socket2268 6759 driver48d
SOCKET_NETWORK socket2269 6760 driver48a
SOCKET_NETWORK socket2270 6761 driver48b
SOCKET_NETWORK socket2271 6762 driver48c
SOCKET_NETWORK socket2272 6763 driver48d
SOCKET_NETWORK socket2273 6764 driver48a
SOCKET_NETWORK socket2274 6765 driver48b
SOCKET_NETWORK socket2275 6766 driver48c
SOCKET_NETWORK socket2276 6767 driver48d
SOCKET_NETWORK socket2277 6768 driver48a
SOCKET_NETWORK socket2278 6769 driver48b
SOCKET_NETWORK socket2279 6770 driver48c
SOCKET_NETWORK socket2280 6771 driver48d
SOCKET_NETWORK socket2281 6772 driver48a
SOCKET_NETWORK socket2282 6773 driver48b
SOCKET_NETWORK socket2283 6774 driver48c
SOCKET_NETWORK socket2284 6775 driver48d
SOCKET_NETWORK socket2285 6776 driver48a
SOCKET_NETWORK socket2286 6777 driver48b
SOCKET_NETWORK socket2287 6778 driver48c
SOCKET_NETWORK socket2288 6779 driver48d
SOCKET_NETWORK socket2289 6780 driver48a
SOCKET_NETWORK socket2290 6781 driver48b
SOCKET_NETWORK socket2291 6782 driver48c
SOCKET_NETWORK socket2292 6783 driver48d
SOCKET_NETWORK socket2293 6784 driver48a
SOCKET_NETWORK socket2294 6785 driver48b
SOCKET_NETWORK socket2295 6786 driver48c
SOCKET_NETWORK socket2296 6787 driver48d
SOCKET_NETWORK socket2297 6788 driver48a
SOCKET_NETWORK socket2298 6789 driver48b
SOCKET_NETWORK socket2299 6790 driver48c
SOCKET_NETWORK socket2300 6791 driver48d
SOCKET_NETWORK socket2301 6792 driver48a
SOCKET_NETWORK socket2302 6793 driver48b
SOCKET_NETWORK socket2303 6794 driver48c
SOCKET_NETWORK socket2304 6795 driver48d
#elif MASTER_NUM9
SOCKET_NETWORK socket2305 6700 driver49a
SOCKET_NETWORK socket2306 6701 driver49b
SOCKET_NETWORK socket2307 6702 driver49c
SOCKET_NETWORK socket2308 6703 driver49d
SOCKET_NETWORK socket2309 6704 driver49a
SOCKET_NETWORK socket2310 6705 driver49b
SOCKET_NETWORK socket2311 6706 driver49c
SOCKET_NETWORK socket2312 6707 driver49d
SOCKET_NETWORK socket2313 6708 driver49a
SOCKET_NETWORK socket2314 6709 driver49b
SOCKET_NETWORK socket2315 6710 driver49c
SOCKET_NETWORK socket2316 6711 driver49d
SOCKET_NETWORK socket2317 6712 driver49a
SOCKET_NETWORK socket2318 6713 driver49b
SOCKET_NETWORK socket2319 6714 driver49c
SOCKET_NETWORK socket2320 6715 driver49d
SOCKET_NETWORK socket2321 6716 driver49a
SOCKET_NETWORK socket2322 6717 driver49b
SOCKET_NETWORK socket2323 6718 driver49c
SOCKET_NETWORK socket2324 6719 driver49d
SOCKET_NETWORK socket2325 6720 driver49a
SOCKET_NETWORK socket2326 6721 driver49b
SOCKET_NETWORK socket2327 6722 driver49c
SOCKET_NETWORK socket2328 6723 driver49d
SOCKET_NETWORK socket2329 6724 driver49a
SOCKET_NETWORK socket2330 6725 driver49b
SOCKET_NETWORK socket2331 6726 driver49c
SOCKET_NETWORK socket2332 6727 driver49d
SOCKET_NETWORK socket2333 6728 driver49a
SOCKET_NETWORK socket2334 6729 driver49b
SOCKET_NETWORK socket2335 6730 driver49c
SOCKET_NETWORK socket2336 6731 driver49d
SOCKET_NETWORK socket2337 6732 driver49a
SOCKET_NETWORK socket2338 6733 driver49b
SOCKET_NETWORK socket2339 6734 driver49c
SOCKET_NETWORK socket2340 6735 driver49d
SOCKET_NETWORK socket2341 6736 driver49a
SOCKET_NETWORK socket2342 6737 driver49b
SOCKET_NETWORK socket2343 6738 driver49c
SOCKET_NETWORK socket2344 6739 driver49d
SOCKET_NETWORK socket2345 6740 driver49a
SOCKET_NETWORK socket2346 6741 driver49b
SOCKET_NETWORK socket2347 6742 driver49c
SOCKET_NETWORK socket2348 6743 driver49d
SOCKET_NETWORK socket2349 6744 driver49a
SOCKET_NETWORK socket2350 6745 driver49b
SOCKET_NETWORK socket2351 6746 driver49c
SOCKET_NETWORK socket2352 6747 driver49d
SOCKET_NETWORK socket2353 6748 driver50a
SOCKET_NETWORK socket2354 6749 driver50b
SOCKET_NETWORK socket2355 6750 driver50c
SOCKET_NETWORK socket2356 6751 driver50d
SOCKET_NETWORK socket2357 6752 driver50a
SOCKET_NETWORK socket2358 6753 driver50b
SOCKET_NETWORK socket2359 6754 driver50c
SOCKET_NETWORK socket2360 6755 driver50d
SOCKET_NETWORK socket2361 6756 driver50a
SOCKET_NETWORK socket2362 6757 driver50b
SOCKET_NETWORK socket2363 6758 driver50c
SOCKET_NETWORK socket2364 6759 driver50d
SOCKET_NETWORK socket2365 6760 driver50a
SOCKET_NETWORK socket2366 6761 driver50b
SOCKET_NETWORK socket2367 6762 driver50c
SOCKET_NETWORK socket2368 6763 driver50d
SOCKET_NETWORK socket2369 6764 driver50a
SOCKET_NETWORK socket2370 6765 driver50b
SOCKET_NETWORK socket2371 6766 driver50c
SOCKET_NETWORK socket2372 6767 driver50d
SOCKET_NETWORK socket2373 6768 driver50a
SOCKET_NETWORK socket2374 6769 driver50b
SOCKET_NETWORK socket2375 6770 driver50c
SOCKET_NETWORK socket2376 6771 driver50d
SOCKET_NETWORK socket2377 6772 driver50a
SOCKET_NETWORK socket2378 6773 driver50b
SOCKET_NETWORK socket2379 6774 driver50c
SOCKET_NETWORK socket2380 6775 driver50d
SOCKET_NETWORK socket2381 6776 driver50a
SOCKET_NETWORK socket2382 6777 driver50b
SOCKET_NETWORK socket2383 6778 driver50c
SOCKET_NETWORK socket2384 6779 driver50d
SOCKET_NETWORK socket2385 6780 driver50a
SOCKET_NETWORK socket2386 6781 driver50b
SOCKET_NETWORK socket2387 6782 driver50c
SOCKET_NETWORK socket2388 6783 driver50d
SOCKET_NETWORK socket2389 6784 driver50a
SOCKET_NETWORK socket2390 6785 driver50b
SOCKET_NETWORK socket2391 6786 driver50c
SOCKET_NETWORK socket2392 6787 driver50d
SOCKET_NETWORK socket2393 6788 driver50a
SOCKET_NETWORK socket2394 6789 driver50b
SOCKET_NETWORK socket2395 6790 driver50c
SOCKET_NETWORK socket2396 6791 driver50d
SOCKET_NETWORK socket2397 6792 driver50a
SOCKET_NETWORK socket2398 6793 driver50b
SOCKET_NETWORK socket2399 6794 driver50c
SOCKET_NETWORK socket2400 6795 driver50d
SOCKET_NETWORK socket2401 6700 driver51a
SOCKET_NETWORK socket2402 6701 driver51b
SOCKET_NETWORK socket2403 6702 driver51c
SOCKET_NETWORK socket2404 6703 driver51d
SOCKET_NETWORK socket2405 6704 driver51a
SOCKET_NETWORK socket2406 6705 driver51b
SOCKET_NETWORK socket2407 6706 driver51c
SOCKET_NETWORK socket2408 6707 driver51d
SOCKET_NETWORK socket2409 6708 driver51a
SOCKET_NETWORK socket2410 6709 driver51b
SOCKET_NETWORK socket2411 6710 driver51c
SOCKET_NETWORK socket2412 6711 driver51d
SOCKET_NETWORK socket2413 6712 driver51a
SOCKET_NETWORK socket2414 6713 driver51b
SOCKET_NETWORK socket2415 6714 driver51c
SOCKET_NETWORK socket2416 6715 driver51d
SOCKET_NETWORK socket2417 6716 driver51a
SOCKET_NETWORK socket2418 6717 driver51b
SOCKET_NETWORK socket2419 6718 driver51c
SOCKET_NETWORK socket2420 6719 driver51d
SOCKET_NETWORK socket2421 6720 driver51a
SOCKET_NETWORK socket2422 6721 driver51b
SOCKET_NETWORK socket2423 6722 driver51c
SOCKET_NETWORK socket2424 6723 driver51d
SOCKET_NETWORK socket2425 6724 driver51a
SOCKET_NETWORK socket2426 6725 driver51b
SOCKET_NETWORK socket2427 6726 driver51c
SOCKET_NETWORK socket2428 6727 driver51d
SOCKET_NETWORK socket2429 6728 driver51a
SOCKET_NETWORK socket2430 6729 driver51b
SOCKET_NETWORK socket2431 6730 driver51c
SOCKET_NETWORK socket2432 6731 driver51d
SOCKET_NETWORK socket2433 6732 driver51a
SOCKET_NETWORK socket2434 6733 driver51b
SOCKET_NETWORK socket2435 6734 driver51c
SOCKET_NETWORK socket2436 6735 driver51d
SOCKET_NETWORK socket2437 6736 driver51a
SOCKET_NETWORK socket2438 6737 driver51b
SOCKET_NETWORK socket2439 6738 driver51c
SOCKET_NETWORK socket2440 6739 driver51d
SOCKET_NETWORK socket2441 6740 driver51a
SOCKET_NETWORK socket2442 6741 driver51b
SOCKET_NETWORK socket2443 6742 driver51c
SOCKET_NETWORK socket2444 6743 driver51d
SOCKET_NETWORK socket2445 6744 driver51a
SOCKET_NETWORK socket2446 6745 driver51b
SOCKET_NETWORK socket2447 6746 driver51c
SOCKET_NETWORK socket2448 6747 driver51d
SOCKET_NETWORK socket2449 6748 driver52a
SOCKET_NETWORK socket2450 6749 driver52b
SOCKET_NETWORK socket2451 6750 driver52c
SOCKET_NETWORK socket2452 6751 driver52d
SOCKET_NETWORK socket2453 6752 driver52a
SOCKET_NETWORK socket2454 6753 driver52b
SOCKET_NETWORK socket2455 6754 driver52c
SOCKET_NETWORK socket2456 6755 driver52d
SOCKET_NETWORK socket2457 6756 driver52a
SOCKET_NETWORK socket2458 6757 driver52b

```
SOCKET_NETWORK socket2459  6758 driver52c
SOCKET_NETWORK socket2460  6759 driver52d
SOCKET_NETWORK socket2461  6760 driver52a
SOCKET_NETWORK socket2462  6761 driver52b
SOCKET_NETWORK socket2463  6762 driver52c
SOCKET_NETWORK socket2464  6763 driver52d
SOCKET_NETWORK socket2465  6764 driver52a
SOCKET_NETWORK socket2466  6765 driver52b
SOCKET_NETWORK socket2467  6766 driver52c
SOCKET_NETWORK socket2468  6767 driver52d
SOCKET_NETWORK socket2469  6768 driver52a
SOCKET_NETWORK socket2470  6769 driver52b
SOCKET_NETWORK socket2471  6770 driver52c
SOCKET_NETWORK socket2472  6771 driver52d
SOCKET_NETWORK socket2473  6772 driver52a
SOCKET_NETWORK socket2474  6773 driver52b
SOCKET_NETWORK socket2475  6774 driver52c
SOCKET_NETWORK socket2476  6775 driver52d
SOCKET_NETWORK socket2477  6776 driver52a
SOCKET_NETWORK socket2478  6777 driver52b
SOCKET_NETWORK socket2479  6778 driver52c
SOCKET_NETWORK socket2480  6779 driver52d
SOCKET_NETWORK socket2481  6780 driver52a
SOCKET_NETWORK socket2482  6781 driver52b
SOCKET_NETWORK socket2483  6782 driver52c
SOCKET_NETWORK socket2484  6783 driver52d
SOCKET_NETWORK socket2485  6784 driver52a
SOCKET_NETWORK socket2486  6785 driver52b
SOCKET_NETWORK socket2487  6786 driver52c
SOCKET_NETWORK socket2488  6787 driver52d
SOCKET_NETWORK socket2489  6788 driver52a
SOCKET_NETWORK socket2490  6789 driver52b
SOCKET_NETWORK socket2491  6790 driver52c
SOCKET_NETWORK socket2492  6791 driver52d
SOCKET_NETWORK socket2493  6792 driver52a
SOCKET_NETWORK socket2494  6793 driver52b
SOCKET_NETWORK socket2495  6794 driver52c
SOCKET_NETWORK socket2496  6795 driver52d
SOCKET_NETWORK socket2497  6700 driver53a
SOCKET_NETWORK socket2498  6701 driver53b
SOCKET_NETWORK socket2499  6702 driver53c
SOCKET_NETWORK socket2500  6703 driver53d
SOCKET_NETWORK socket2501  6704 driver53a
SOCKET_NETWORK socket2502  6705 driver53b
SOCKET_NETWORK socket2503  6706 driver53c
SOCKET_NETWORK socket2504  6707 driver53d
SOCKET_NETWORK socket2505  6708 driver53a
SOCKET_NETWORK socket2506  6709 driver53b
SOCKET_NETWORK socket2507  6710 driver53c
SOCKET_NETWORK socket2508  6711 driver53d
SOCKET_NETWORK socket2509  6712 driver53a
SOCKET_NETWORK socket2510  6713 driver53b
SOCKET_NETWORK socket2511  6714 driver53c
SOCKET_NETWORK socket2512  6715 driver53d
SOCKET_NETWORK socket2513  6716 driver53a
SOCKET_NETWORK socket2514  6717 driver53b
SOCKET_NETWORK socket2515  6718 driver53c
SOCKET_NETWORK socket2516  6719 driver53d
SOCKET_NETWORK socket2517  6720 driver53a
SOCKET_NETWORK socket2518  6721 driver53b
SOCKET_NETWORK socket2519  6722 driver53c
SOCKET_NETWORK socket2520  6723 driver53d
SOCKET_NETWORK socket2521  6724 driver53a
SOCKET_NETWORK socket2522  6725 driver53b
SOCKET_NETWORK socket2523  6726 driver53c
SOCKET_NETWORK socket2524  6727 driver53d
SOCKET_NETWORK socket2525  6728 driver53a
SOCKET_NETWORK socket2526  6729 driver53b
SOCKET_NETWORK socket2527  6730 driver53c
SOCKET_NETWORK socket2528  6731 driver53d
SOCKET_NETWORK socket2529  6732 driver53a
SOCKET_NETWORK socket2530  6733 driver53b
SOCKET_NETWORK socket2531  6734 driver53c
SOCKET_NETWORK socket2532  6735 driver53d
SOCKET_NETWORK socket2533  6736 driver53a
SOCKET_NETWORK socket2534  6737 driver53b
SOCKET_NETWORK socket2535  6738 driver53c
SOCKET_NETWORK socket2536  6739 driver53d
SOCKET_NETWORK socket2537  6740 driver53a
SOCKET_NETWORK socket2538  6741 driver53b
SOCKET_NETWORK socket2539  6742 driver53c
SOCKET_NETWORK socket2540  6743 driver53d
SOCKET_NETWORK socket2541  6744 driver53a
SOCKET_NETWORK socket2542  6745 driver53b
SOCKET_NETWORK socket2543  6746 driver53c
SOCKET_NETWORK socket2544  6747 driver53d
SOCKET_NETWORK socket2545  6748 driver54a
SOCKET_NETWORK socket2546  6749 driver54b
SOCKET_NETWORK socket2547  6750 driver54c
SOCKET_NETWORK socket2548  6751 driver54d
SOCKET_NETWORK socket2549  6752 driver54a
SOCKET_NETWORK socket2550  6753 driver54b
SOCKET_NETWORK socket2551  6754 driver54c
SOCKET_NETWORK socket2552  6755 driver54d
SOCKET_NETWORK socket2553  6756 driver54a
SOCKET_NETWORK socket2554  6757 driver54b
SOCKET_NETWORK socket2555  6758 driver54c
SOCKET_NETWORK socket2556  6759 driver54d
SOCKET_NETWORK socket2557  6760 driver54a
SOCKET_NETWORK socket2558  6761 driver54b
SOCKET_NETWORK socket2559  6762 driver54c
SOCKET_NETWORK socket2560  6763 driver54d
SOCKET_NETWORK socket2561  6764 driver54a
SOCKET_NETWORK socket2562  6765 driver54b
SOCKET_NETWORK socket2563  6766 driver54c
SOCKET_NETWORK socket2564  6767 driver54d
SOCKET_NETWORK socket2565  6768 driver54a
SOCKET_NETWORK socket2566  6769 driver54b
SOCKET_NETWORK socket2567  6770 driver54c
SOCKET_NETWORK socket2568  6771 driver54d
SOCKET_NETWORK socket2569  6772 driver54a
SOCKET_NETWORK socket2570  6773 driver54b
SOCKET_NETWORK socket2571  6774 driver54c
SOCKET_NETWORK socket2572  6775 driver54d
SOCKET_NETWORK socket2573  6776 driver54a
SOCKET_NETWORK socket2574  6777 driver54b
SOCKET_NETWORK socket2575  6778 driver54c
SOCKET_NETWORK socket2576  6779 driver54d
SOCKET_NETWORK socket2577  6780 driver54a
SOCKET_NETWORK socket2578  6781 driver54b
SOCKET_NETWORK socket2579  6782 driver54c
SOCKET_NETWORK socket2580  6783 driver54d
SOCKET_NETWORK socket2581  6784 driver54a
SOCKET_NETWORK socket2582  6785 driver54b
SOCKET_NETWORK socket2583  6786 driver54c
SOCKET_NETWORK socket2584  6787 driver54d
SOCKET_NETWORK socket2585  6788 driver54a
SOCKET_NETWORK socket2586  6789 driver54b
SOCKET_NETWORK socket2587  6790 driver54c
SOCKET_NETWORK socket2588  6791 driver54d
SOCKET_NETWORK socket2589  6792 driver54a
SOCKET_NETWORK socket2590  6793 driver54b
SOCKET_NETWORK socket2591  6794 driver54c
SOCKET_NETWORK socket2592  6795 driver54d
#elif MASTER_NUM10
SOCKET_NETWORK socket2593  6700 driver55a
SOCKET_NETWORK socket2594  6701 driver55b
SOCKET_NETWORK socket2595  6702 driver55c
SOCKET_NETWORK socket2596  6703 driver55d
SOCKET_NETWORK socket2597  6704 driver55a
SOCKET_NETWORK socket2598  6705 driver55b
SOCKET_NETWORK socket2599  6706 driver55c
SOCKET_NETWORK socket2600  6707 driver55d
SOCKET_NETWORK socket2601  6708 driver55a
SOCKET_NETWORK socket2602  6709 driver55b
SOCKET_NETWORK socket2603  6710 driver55c
SOCKET_NETWORK socket2604  6711 driver55d
SOCKET_NETWORK socket2605  6712 driver55a
SOCKET_NETWORK socket2606  6713 driver55b
SOCKET_NETWORK socket2607  6714 driver55c
SOCKET_NETWORK socket2608  6715 driver55d
SOCKET_NETWORK socket2609  6716 driver55a
SOCKET_NETWORK socket2610  6717 driver55b
SOCKET_NETWORK socket2611  6718 driver55c
SOCKET_NETWORK socket2612  6719 driver55d
SOCKET_NETWORK socket2613  6720 driver55a
SOCKET_NETWORK socket2614  6721 driver55b
SOCKET_NETWORK socket2615  6722 driver55c
SOCKET_NETWORK socket2616  6723 driver55d
SOCKET_NETWORK socket2617  6724 driver55a
SOCKET_NETWORK socket2618  6725 driver55b
SOCKET_NETWORK socket2619  6726 driver55c
SOCKET_NETWORK socket2620  6727 driver55d
SOCKET_NETWORK socket2621  6728 driver55a
SOCKET_NETWORK socket2622  6729 driver55b
SOCKET_NETWORK socket2623  6730 driver55c
SOCKET_NETWORK socket2624  6731 driver55d
SOCKET_NETWORK socket2625  6732 driver55a
SOCKET_NETWORK socket2626  6733 driver55b
SOCKET_NETWORK socket2627  6734 driver55c
SOCKET_NETWORK socket2628  6735 driver55d
SOCKET_NETWORK socket2629  6736 driver55a
SOCKET_NETWORK socket2630  6737 driver55b
SOCKET_NETWORK socket2631  6738 driver55c
SOCKET_NETWORK socket2632  6739 driver55d
SOCKET_NETWORK socket2633  6740 driver55a
SOCKET_NETWORK socket2634  6741 driver55b
SOCKET_NETWORK socket2635  6742 driver55c
SOCKET_NETWORK socket2636  6743 driver55d
SOCKET_NETWORK socket2637  6744 driver55a
SOCKET_NETWORK socket2638  6745 driver55b
SOCKET_NETWORK socket2639  6746 driver55c
SOCKET_NETWORK socket2640  6747 driver55d
SOCKET_NETWORK socket2641  6748 driver56a
SOCKET_NETWORK socket2642  6749 driver56b
SOCKET_NETWORK socket2643  6750 driver56c
SOCKET_NETWORK socket2644  6751 driver56d
SOCKET_NETWORK socket2645  6752 driver56a
SOCKET_NETWORK socket2646  6753 driver56b
SOCKET_NETWORK socket2647  6754 driver56c
SOCKET_NETWORK socket2648  6755 driver56d
SOCKET_NETWORK socket2649  6756 driver56a
SOCKET_NETWORK socket2650  6757 driver56b
SOCKET_NETWORK socket2651  6758 driver56c
SOCKET_NETWORK socket2652  6759 driver56d
SOCKET_NETWORK socket2653  6760 driver56a
SOCKET_NETWORK socket2654  6761 driver56b
SOCKET_NETWORK socket2655  6762 driver56c
```

SOCKET_NETWORK socket2656 6763 driver56d
SOCKET_NETWORK socket2657 6764 driver56a
SOCKET_NETWORK socket2658 6765 driver56b
SOCKET_NETWORK socket2659 6766 driver56c
SOCKET_NETWORK socket2660 6767 driver56d
SOCKET_NETWORK socket2661 6768 driver56a
SOCKET_NETWORK socket2662 6769 driver56b
SOCKET_NETWORK socket2663 6770 driver56c
SOCKET_NETWORK socket2664 6771 driver56d
SOCKET_NETWORK socket2665 6772 driver56a
SOCKET_NETWORK socket2666 6773 driver56b
SOCKET_NETWORK socket2667 6774 driver56c
SOCKET_NETWORK socket2668 6775 driver56d
SOCKET_NETWORK socket2669 6776 driver56a
SOCKET_NETWORK socket2670 6777 driver56b
SOCKET_NETWORK socket2671 6778 driver56c
SOCKET_NETWORK socket2672 6779 driver56d
SOCKET_NETWORK socket2673 6780 driver56a
SOCKET_NETWORK socket2674 6781 driver56b
SOCKET_NETWORK socket2675 6782 driver56c
SOCKET_NETWORK socket2676 6783 driver56d
SOCKET_NETWORK socket2677 6784 driver56a
SOCKET_NETWORK socket2678 6785 driver56b
SOCKET_NETWORK socket2679 6786 driver56c
SOCKET_NETWORK socket2680 6787 driver56d
SOCKET_NETWORK socket2681 6788 driver56a
SOCKET_NETWORK socket2682 6789 driver56b
SOCKET_NETWORK socket2683 6790 driver56c
SOCKET_NETWORK socket2684 6791 driver56d
SOCKET_NETWORK socket2685 6792 driver56a
SOCKET_NETWORK socket2686 6793 driver56b
SOCKET_NETWORK socket2687 6794 driver56c
SOCKET_NETWORK socket2688 6795 driver56d
SOCKET_NETWORK socket2689 6700 driver57a
SOCKET_NETWORK socket2690 6701 driver57b
SOCKET_NETWORK socket2691 6702 driver57c
SOCKET_NETWORK socket2692 6703 driver57d
SOCKET_NETWORK socket2693 6704 driver57a
SOCKET_NETWORK socket2694 6705 driver57b
SOCKET_NETWORK socket2695 6706 driver57c
SOCKET_NETWORK socket2696 6707 driver57d
SOCKET_NETWORK socket2697 6708 driver57a
SOCKET_NETWORK socket2698 6709 driver57b
SOCKET_NETWORK socket2699 6710 driver57c
SOCKET_NETWORK socket2700 6711 driver57d
SOCKET_NETWORK socket2701 6712 driver57a
SOCKET_NETWORK socket2702 6713 driver57b
SOCKET_NETWORK socket2703 6714 driver57c
SOCKET_NETWORK socket2704 6715 driver57d
SOCKET_NETWORK socket2705 6716 driver57a
SOCKET_NETWORK socket2706 6717 driver57b
SOCKET_NETWORK socket2707 6718 driver57c
SOCKET_NETWORK socket2708 6719 driver57d
SOCKET_NETWORK socket2709 6720 driver57a
SOCKET_NETWORK socket2710 6721 driver57b
SOCKET_NETWORK socket2711 6722 driver57c
SOCKET_NETWORK socket2712 6723 driver57d
SOCKET_NETWORK socket2713 6724 driver57a
SOCKET_NETWORK socket2714 6725 driver57b
SOCKET_NETWORK socket2715 6726 driver57c
SOCKET_NETWORK socket2716 6727 driver57d
SOCKET_NETWORK socket2717 6728 driver57a
SOCKET_NETWORK socket2718 6729 driver57b
SOCKET_NETWORK socket2719 6730 driver57c
SOCKET_NETWORK socket2720 6731 driver57d
SOCKET_NETWORK socket2721 6732 driver57a
SOCKET_NETWORK socket2722 6733 driver57b
SOCKET_NETWORK socket2723 6734 driver57c
SOCKET_NETWORK socket2724 6735 driver57d
SOCKET_NETWORK socket2725 6736 driver57a
SOCKET_NETWORK socket2726 6737 driver57b
SOCKET_NETWORK socket2727 6738 driver57c
SOCKET_NETWORK socket2728 6739 driver57d
SOCKET_NETWORK socket2729 6740 driver57a
SOCKET_NETWORK socket2730 6741 driver57b
SOCKET_NETWORK socket2731 6742 driver57c
SOCKET_NETWORK socket2732 6743 driver57d
SOCKET_NETWORK socket2733 6744 driver57a
SOCKET_NETWORK socket2734 6745 driver57b
SOCKET_NETWORK socket2735 6746 driver57c
SOCKET_NETWORK socket2736 6747 driver57d
SOCKET_NETWORK socket2737 6748 driver58a
SOCKET_NETWORK socket2738 6749 driver58b
SOCKET_NETWORK socket2739 6750 driver58c
SOCKET_NETWORK socket2740 6751 driver58d
SOCKET_NETWORK socket2741 6752 driver58a
SOCKET_NETWORK socket2742 6753 driver58b
SOCKET_NETWORK socket2743 6754 driver58c
SOCKET_NETWORK socket2744 6755 driver58d
SOCKET_NETWORK socket2745 6756 driver58a
SOCKET_NETWORK socket2746 6757 driver58b
SOCKET_NETWORK socket2747 6758 driver58c
SOCKET_NETWORK socket2748 6759 driver58d
SOCKET_NETWORK socket2749 6760 driver58a
SOCKET_NETWORK socket2750 6761 driver58b
SOCKET_NETWORK socket2751 6762 driver58c
SOCKET_NETWORK socket2752 6763 driver58d
SOCKET_NETWORK socket2753 6764 driver58a
SOCKET_NETWORK socket2754 6765 driver58b
SOCKET_NETWORK socket2755 6766 driver58c
SOCKET_NETWORK socket2756 6767 driver58d
SOCKET_NETWORK socket2757 6768 driver58a
SOCKET_NETWORK socket2758 6769 driver58b
SOCKET_NETWORK socket2759 6770 driver58c
SOCKET_NETWORK socket2760 6771 driver58d
SOCKET_NETWORK socket2761 6772 driver58a
SOCKET_NETWORK socket2762 6773 driver58b
SOCKET_NETWORK socket2763 6774 driver58c
SOCKET_NETWORK socket2764 6775 driver58d
SOCKET_NETWORK socket2765 6776 driver58a
SOCKET_NETWORK socket2766 6777 driver58b
SOCKET_NETWORK socket2767 6778 driver58c
SOCKET_NETWORK socket2768 6779 driver58d
SOCKET_NETWORK socket2769 6780 driver58a
SOCKET_NETWORK socket2770 6781 driver58b
SOCKET_NETWORK socket2771 6782 driver58c
SOCKET_NETWORK socket2772 6783 driver58d
SOCKET_NETWORK socket2773 6784 driver58a
SOCKET_NETWORK socket2774 6785 driver58b
SOCKET_NETWORK socket2775 6786 driver58c
SOCKET_NETWORK socket2776 6787 driver58d
SOCKET_NETWORK socket2777 6788 driver58a
SOCKET_NETWORK socket2778 6789 driver58b
SOCKET_NETWORK socket2779 6790 driver58c
SOCKET_NETWORK socket2780 6791 driver58d
SOCKET_NETWORK socket2781 6792 driver58a
SOCKET_NETWORK socket2782 6793 driver58b
SOCKET_NETWORK socket2783 6794 driver58c
SOCKET_NETWORK socket2784 6795 driver58d
SOCKET_NETWORK socket2785 6700 driver59a
SOCKET_NETWORK socket2786 6701 driver59b
SOCKET_NETWORK socket2787 6702 driver59c
SOCKET_NETWORK socket2788 6703 driver59d
SOCKET_NETWORK socket2789 6704 driver59a
SOCKET_NETWORK socket2790 6705 driver59b
SOCKET_NETWORK socket2791 6706 driver59c
SOCKET_NETWORK socket2792 6707 driver59d
SOCKET_NETWORK socket2793 6708 driver59a
SOCKET_NETWORK socket2794 6709 driver59b
SOCKET_NETWORK socket2795 6710 driver59c
SOCKET_NETWORK socket2796 6711 driver59d
SOCKET_NETWORK socket2797 6712 driver59a
SOCKET_NETWORK socket2798 6713 driver59b
SOCKET_NETWORK socket2799 6714 driver59c
SOCKET_NETWORK socket2800 6715 driver59d
SOCKET_NETWORK socket2801 6716 driver59a
SOCKET_NETWORK socket2802 6717 driver59b
SOCKET_NETWORK socket2803 6718 driver59c
SOCKET_NETWORK socket2804 6719 driver59d
SOCKET_NETWORK socket2805 6720 driver59a
SOCKET_NETWORK socket2806 6721 driver59b
SOCKET_NETWORK socket2807 6722 driver59c
SOCKET_NETWORK socket2808 6723 driver59d
SOCKET_NETWORK socket2809 6724 driver59a
SOCKET_NETWORK socket2810 6725 driver59b
SOCKET_NETWORK socket2811 6726 driver59c
SOCKET_NETWORK socket2812 6727 driver59d
SOCKET_NETWORK socket2813 6728 driver59a
SOCKET_NETWORK socket2814 6729 driver59b
SOCKET_NETWORK socket2815 6730 driver59c
SOCKET_NETWORK socket2816 6731 driver59d
SOCKET_NETWORK socket2817 6732 driver59a
SOCKET_NETWORK socket2818 6733 driver59b
SOCKET_NETWORK socket2819 6734 driver59c
SOCKET_NETWORK socket2820 6735 driver59d
SOCKET_NETWORK socket2821 6736 driver59a
SOCKET_NETWORK socket2822 6737 driver59b
SOCKET_NETWORK socket2823 6738 driver59c
SOCKET_NETWORK socket2824 6739 driver59d
SOCKET_NETWORK socket2825 6740 driver59a
SOCKET_NETWORK socket2826 6741 driver59b
SOCKET_NETWORK socket2827 6742 driver59c
SOCKET_NETWORK socket2828 6743 driver59d
SOCKET_NETWORK socket2829 6744 driver59a
SOCKET_NETWORK socket2830 6745 driver59b
SOCKET_NETWORK socket2831 6746 driver59c
SOCKET_NETWORK socket2832 6747 driver59d
SOCKET_NETWORK socket2833 6748 driver60a
SOCKET_NETWORK socket2834 6749 driver60b
SOCKET_NETWORK socket2835 6750 driver60c
SOCKET_NETWORK socket2836 6751 driver60d
SOCKET_NETWORK socket2837 6752 driver60a
SOCKET_NETWORK socket2838 6753 driver60b
SOCKET_NETWORK socket2839 6754 driver60c
SOCKET_NETWORK socket2840 6755 driver60d
SOCKET_NETWORK socket2841 6756 driver60a
SOCKET_NETWORK socket2842 6757 driver60b
SOCKET_NETWORK socket2843 6758 driver60c
SOCKET_NETWORK socket2844 6759 driver60d
SOCKET_NETWORK socket2845 6760 driver60a
SOCKET_NETWORK socket2846 6761 driver60b
SOCKET_NETWORK socket2847 6762 driver60c
SOCKET_NETWORK socket2848 6763 driver60d
SOCKET_NETWORK socket2849 6764 driver60a
SOCKET_NETWORK socket2850 6765 driver60b
SOCKET_NETWORK socket2851 6766 driver60c
SOCKET_NETWORK socket2852 6767 driver60d
SOCKET_NETWORK socket2853 6768 driver60a

```
SOCKET_NETWORK socket2854   6769 driver60b
SOCKET_NETWORK socket2855   6770 driver60c
SOCKET_NETWORK socket2856   6771 driver60d
SOCKET_NETWORK socket2857   6772 driver60a
SOCKET_NETWORK socket2858   6773 driver60b
SOCKET_NETWORK socket2859   6774 driver60c
SOCKET_NETWORK socket2860   6775 driver60d
SOCKET_NETWORK socket2861   6776 driver60a
SOCKET_NETWORK socket2862   6777 driver60b
SOCKET_NETWORK socket2863   6778 driver60c
SOCKET_NETWORK socket2864   6779 driver60d
SOCKET_NETWORK socket2865   6780 driver60a
SOCKET_NETWORK socket2866   6781 driver60b
SOCKET_NETWORK socket2867   6782 driver60c
SOCKET_NETWORK socket2868   6783 driver60d
SOCKET_NETWORK socket2869   6784 driver60a
SOCKET_NETWORK socket2870   6785 driver60b
SOCKET_NETWORK socket2871   6786 driver60c
SOCKET_NETWORK socket2872   6787 driver60d
SOCKET_NETWORK socket2873   6788 driver60a
SOCKET_NETWORK socket2874   6789 driver60b
SOCKET_NETWORK socket2875   6790 driver60c
SOCKET_NETWORK socket2876   6791 driver60d
SOCKET_NETWORK socket2877   6792 driver60a
SOCKET_NETWORK socket2878   6793 driver60b
SOCKET_NETWORK socket2879   6794 driver60c
SOCKET_NETWORK socket2880   6795 driver60d
#endif
/*-----------------------------------------------------------------*/
OUTPUTNAME="regattaH"
CPU=48
#if 0
BEGIN_WAIT=5:00
RAMPUP=42:30
RUNTIME=30:00
RAMPDOWN_WAIT=5:00
RAMPDOWN=17:00
#else
BEGIN_WAIT=15:00
RAMPUP=25:00
RUNTIME=10:00
RAMPDOWN_WAIT=5:00
RAMPDOWN=17:00
#endif
INTERVAL=1:00 /* Interval to calculate mix from*/
LOGIN_MAX_LOAD = 2
LOGIN_BEGIN = 0 /* skip login state if set to 1 */
NOBEGIN = 1
KEYSTROKE_PACKET_SIZE = 0
MAX_CONCURRENT_SPAWN = 7
SPAWN_COUNT = 2
MIN_PORT = 8088
MAX_PORT = 8089
/* User variables. Think, Emulex Delay, %desired, %min, %max */
#if 1 /* Testing */
NEWORDER = "12.02, 0, 0"
PAYMENT = "12.02, 0, 0, 43.03, 43.03, 43.03 "
ORDSTAT = "10.01, 0, 0, 4.02, 4.02, 4.02 "
DELIVERY = "05.02, 0, 0, 4.02, 4.02, 4.02 "
STOCKLEV = "05.02, 0, 0, 4.02, 4.02, 4.02 "
#elif 0 /* From rteparams.null */
NEWORDER = "12.25, 0.42, 0.38"
PAYMENT = "12.25, 0.19, 0.23, 43.2, 41.1, 45.3 "
ORDSTAT = "10.50, 0.39, 0.21, 4.1, 3.9, 4.3 "
DELIVERY = "05.5, 0.19, 0.15, 4.1, 3.9, 4.3 "
STOCKLEV = "05.5, 0.25, 0.18, 4.1, 3.9, 4.3 "
#elif 0 /* From Pookeepsie */
NEWORDER = "16.25, 0.42, 0.38"
PAYMENT = "16.25, 0.19, 0.23, 43.15, 43.15, 43.15 "
ORDSTAT = "14.50, 0.39, 0.21, 4.03, 4.03, 4.03 "
DELIVERY = "09.50, 0.19, 0.15, 4.03, 4.03, 4.03 "
STOCKLEV = "09.50, 0.25, 0.18, 4.03, 4.03, 4.03 "
#endif
/*---- Starting users on sockets ---------------------------------------*/
#if MASTER_NUM1
START_RANGE client11a socket1 210 0-21
START_RANGE client11b socket2 210 21-42
START_RANGE client11c socket3 210 42-63
START_RANGE client11d socket4 210 63-84
START_RANGE client11e socket5 210 84-105
START_RANGE client11f socket6 210 105-126
START_RANGE client11g socket7 210 126-147
START_RANGE client11h socket8 210 147-168
START_RANGE client11i socket9 210 168-189
START_RANGE client11j socket10 220 189-211
START_RANGE client11k socket11 210 211-232
START_RANGE client11l socket12 210 232-253
START_RANGE client11a socket13 210 253-274
START_RANGE client11b socket14 210 274-295
START_RANGE client11c socket15 210 295-316
START_RANGE client11d socket16 210 316-337
START_RANGE client11e socket17 210 337-358
START_RANGE client11f socket18 210 358-379
START_RANGE client11g socket19 210 379-400
START_RANGE client11h socket20 220 400-422
START_RANGE client11i socket21 210 422-443
START_RANGE client11j socket22 210 443-464
START_RANGE client11k socket23 210 464-485
START_RANGE client11l socket24 210 485-506
START_RANGE client11a socket25 210 506-527
START_RANGE client11b socket26 210 527-548
START_RANGE client11c socket27 210 548-569
START_RANGE client11d socket28 210 569-590
START_RANGE client11e socket29 210 590-611
START_RANGE client11f socket30 220 611-633
START_RANGE client11g socket31 210 633-654
START_RANGE client11h socket32 210 654-675
START_RANGE client11i socket33 210 675-696
START_RANGE client11j socket34 210 696-717
START_RANGE client11k socket35 210 717-738
START_RANGE client11l socket36 210 738-759
START_RANGE client11a socket37 210 759-780
START_RANGE client11b socket38 210 780-801
START_RANGE client11c socket39 210 801-822
START_RANGE client11d socket40 220 822-844
START_RANGE client11e socket41 210 844-865
START_RANGE client11f socket42 210 865-886
START_RANGE client11g socket43 210 886-907
START_RANGE client11h socket44 210 907-928
START_RANGE client11i socket45 210 928-949
START_RANGE client11j socket46 210 949-970
START_RANGE client11k socket47 210 970-991
START_RANGE client11l socket48 210 991-1012

START_RANGE client12a socket49 210 1012-1033
START_RANGE client12b socket50 220 1033-1055
START_RANGE client12c socket51 210 1055-1076
START_RANGE client12d socket52 210 1076-1097
START_RANGE client12e socket53 210 1097-1118
START_RANGE client12f socket54 210 1118-1139
START_RANGE client12g socket55 210 1139-1160
START_RANGE client12h socket56 210 1160-1181
START_RANGE client12i socket57 210 1181-1202
START_RANGE client12j socket58 210 1202-1223
START_RANGE client12k socket59 210 1223-1244
START_RANGE client12l socket60 220 1244-1266
START_RANGE client12a socket61 210 1266-1287
START_RANGE client12b socket62 210 1287-1308
START_RANGE client12c socket63 210 1308-1329
START_RANGE client12d socket64 210 1329-1350
START_RANGE client12e socket65 210 1350-1371
START_RANGE client12f socket66 210 1371-1392
START_RANGE client12g socket67 210 1392-1413
START_RANGE client12h socket68 210 1413-1434
START_RANGE client12i socket69 210 1434-1455
START_RANGE client12j socket70 220 1455-1477
START_RANGE client12k socket71 210 1477-1498
START_RANGE client12l socket72 210 1498-1519
START_RANGE client12a socket73 210 1519-1540
START_RANGE client12b socket74 210 1540-1561
START_RANGE client12c socket75 210 1561-1582
START_RANGE client12d socket76 210 1582-1603
START_RANGE client12e socket77 210 1603-1624
START_RANGE client12f socket78 210 1624-1645
START_RANGE client12g socket79 210 1645-1666
START_RANGE client12h socket80 220 1666-1688
START_RANGE client12i socket81 210 1688-1709
START_RANGE client12j socket82 210 1709-1730
START_RANGE client12k socket83 210 1730-1751
START_RANGE client12l socket84 210 1751-1772
START_RANGE client12a socket85 210 1772-1793
START_RANGE client12b socket86 210 1793-1814
START_RANGE client12c socket87 210 1814-1835
START_RANGE client12d socket88 210 1835-1856
START_RANGE client12e socket89 210 1856-1877
START_RANGE client12f socket90 220 1877-1899
START_RANGE client12g socket91 210 1899-1920
START_RANGE client12h socket92 210 1920-1941
START_RANGE client12i socket93 210 1941-1962
START_RANGE client12j socket94 210 1962-1983
START_RANGE client12k socket95 210 1983-2004
START_RANGE client12l socket96 210 2004-2025

START_RANGE client21a socket97 210 2025-2046
START_RANGE client21b socket98 210 2046-2067
START_RANGE client21c socket99 210 2067-2088
START_RANGE client21d socket100 220 2088-2110
START_RANGE client21e socket101 210 2110-2131
START_RANGE client21f socket102 210 2131-2152
START_RANGE client21g socket103 210 2152-2173
START_RANGE client21h socket104 210 2173-2194
START_RANGE client21i socket105 210 2194-2215
START_RANGE client21j socket106 210 2215-2236
START_RANGE client21k socket107 210 2236-2257
START_RANGE client21l socket108 210 2257-2278
START_RANGE client21a socket109 210 2278-2299
START_RANGE client21b socket110 220 2299-2321
START_RANGE client21c socket111 210 2321-2342
START_RANGE client21d socket112 210 2342-2363
START_RANGE client21e socket113 210 2363-2384
START_RANGE client21f socket114 210 2384-2405
START_RANGE client21g socket115 210 2405-2426
START_RANGE client21h socket116 210 2426-2447
START_RANGE client21i socket117 210 2447-2468
START_RANGE client21j socket118 210 2468-2489
START_RANGE client21k socket119 210 2489-2510
START_RANGE client21l socket120 220 2510-2532
START_RANGE client21a socket121 210 2532-2553
```

```
START_RANGE client21b socket122 210 2553-2574
START_RANGE client21c socket123 210 2574-2595
START_RANGE client21d socket124 210 2595-2616
START_RANGE client21e socket125 210 2616-2637
START_RANGE client21f socket126 210 2637-2658
START_RANGE client21g socket127 210 2658-2679
START_RANGE client21h socket128 210 2679-2700
START_RANGE client21i socket129 210 2700-2721
START_RANGE client21j socket130 220 2721-2743
START_RANGE client21k socket131 210 2743-2764
START_RANGE client21l socket132 210 2764-2785
START_RANGE client21a socket133 210 2785-2806
START_RANGE client21b socket134 210 2806-2827
START_RANGE client21c socket135 210 2827-2848
START_RANGE client21d socket136 210 2848-2869
START_RANGE client21e socket137 210 2869-2890
START_RANGE client21f socket138 210 2890-2911
START_RANGE client21g socket139 210 2911-2932
START_RANGE client21h socket140 220 2932-2954
START_RANGE client21i socket141 210 2954-2975
START_RANGE client21j socket142 210 2975-2996
START_RANGE client21k socket143 210 2996-3017
START_RANGE client21l socket144 210 3017-3038

START_RANGE client22a socket145 210 3038-3059
START_RANGE client22b socket146 210 3059-3080
START_RANGE client22c socket147 210 3080-3101
START_RANGE client22d socket148 210 3101-3122
START_RANGE client22e socket149 210 3122-3143
START_RANGE client22f socket150 220 3143-3165
START_RANGE client22g socket151 210 3165-3186
START_RANGE client22h socket152 210 3186-3207
START_RANGE client22i socket153 210 3207-3228
START_RANGE client22j socket154 210 3228-3249
START_RANGE client22k socket155 210 3249-3270
START_RANGE client22l socket156 210 3270-3291
START_RANGE client22a socket157 210 3291-3312
START_RANGE client22b socket158 210 3312-3333
START_RANGE client22c socket159 210 3333-3354
START_RANGE client22d socket160 220 3354-3376
START_RANGE client22e socket161 210 3376-3397
START_RANGE client22f socket162 210 3397-3418
START_RANGE client22g socket163 210 3418-3439
START_RANGE client22h socket164 210 3439-3460
START_RANGE client22i socket165 210 3460-3481
START_RANGE client22j socket166 210 3481-3502
START_RANGE client22k socket167 210 3502-3523
START_RANGE client22l socket168 210 3523-3544
START_RANGE client22a socket169 210 3544-3565
START_RANGE client22b socket170 220 3565-3587
START_RANGE client22c socket171 210 3587-3608
START_RANGE client22d socket172 210 3608-3629
START_RANGE client22e socket173 210 3629-3650
START_RANGE client22f socket174 210 3650-3671
START_RANGE client22g socket175 210 3671-3692
START_RANGE client22h socket176 210 3692-3713
START_RANGE client22i socket177 210 3713-3734
START_RANGE client22j socket178 210 3734-3755
START_RANGE client22k socket179 210 3755-3776
START_RANGE client22l socket180 220 3776-3798
START_RANGE client22a socket181 210 3798-3819
START_RANGE client22b socket182 210 3819-3840
START_RANGE client22c socket183 210 3840-3861
START_RANGE client22d socket184 210 3861-3882
START_RANGE client22e socket185 210 3882-3903
START_RANGE client22f socket186 210 3903-3924
START_RANGE client22g socket187 210 3924-3945
START_RANGE client22h socket188 210 3945-3966
START_RANGE client22i socket189 210 3966-3987
START_RANGE client22j socket190 220 3987-4009
START_RANGE client22k socket191 210 4009-4030
START_RANGE client22l socket192 210 4030-4051

START_RANGE client31a socket193 210 4051-4072
START_RANGE client31b socket194 210 4072-4093
START_RANGE client31c socket195 210 4093-4114
START_RANGE client31d socket196 210 4114-4135
START_RANGE client31e socket197 210 4135-4156
START_RANGE client31f socket198 210 4156-4177
START_RANGE client31g socket199 210 4177-4198
START_RANGE client31h socket200 220 4198-4220
START_RANGE client31i socket201 210 4220-4241
START_RANGE client31j socket202 210 4241-4262
START_RANGE client31k socket203 210 4262-4283
START_RANGE client31l socket204 210 4283-4304
START_RANGE client31a socket205 210 4304-4325
START_RANGE client31b socket206 210 4325-4346
START_RANGE client31c socket207 210 4346-4367
START_RANGE client31d socket208 210 4367-4388
START_RANGE client31e socket209 210 4388-4409
START_RANGE client31f socket210 220 4409-4431
START_RANGE client31g socket211 210 4431-4452
START_RANGE client31h socket212 210 4452-4473
START_RANGE client31i socket213 210 4473-4494
START_RANGE client31j socket214 210 4494-4515
START_RANGE client31k socket215 210 4515-4536
START_RANGE client31l socket216 210 4536-4557
START_RANGE client31a socket217 210 4557-4578
START_RANGE client31b socket218 210 4578-4599
START_RANGE client31c socket219 210 4599-4620
START_RANGE client31d socket220 220 4620-4642
START_RANGE client31e socket221 210 4642-4663
START_RANGE client31f socket222 210 4663-4684
START_RANGE client31g socket223 210 4684-4705
START_RANGE client31h socket224 210 4705-4726
START_RANGE client31i socket225 210 4726-4747
START_RANGE client31j socket226 210 4747-4768
START_RANGE client31k socket227 210 4768-4789
START_RANGE client31l socket228 210 4789-4810
START_RANGE client31a socket229 210 4810-4831
START_RANGE client31b socket230 220 4831-4853
START_RANGE client31c socket231 210 4853-4874
START_RANGE client31d socket232 210 4874-4895
START_RANGE client31e socket233 210 4895-4916
START_RANGE client31f socket234 210 4916-4937
START_RANGE client31g socket235 210 4937-4958
START_RANGE client31h socket236 210 4958-4979
START_RANGE client31i socket237 210 4979-5000
START_RANGE client31j socket238 210 5000-5021
START_RANGE client31k socket239 210 5021-5042
START_RANGE client31l socket240 220 5042-5064

START_RANGE client32a socket241 210 5064-5085
START_RANGE client32b socket242 210 5085-5106
START_RANGE client32c socket243 210 5106-5127
START_RANGE client32d socket244 210 5127-5148
START_RANGE client32e socket245 210 5148-5169
START_RANGE client32f socket246 210 5169-5190
START_RANGE client32g socket247 210 5190-5211
START_RANGE client32h socket248 210 5211-5232
START_RANGE client32i socket249 210 5232-5253
START_RANGE client32j socket250 220 5253-5275
START_RANGE client32k socket251 210 5275-5296
START_RANGE client32l socket252 210 5296-5317
START_RANGE client32a socket253 210 5317-5338
START_RANGE client32b socket254 210 5338-5359
START_RANGE client32c socket255 210 5359-5380
START_RANGE client32d socket256 210 5380-5401
START_RANGE client32e socket257 210 5401-5422
START_RANGE client32f socket258 210 5422-5443
START_RANGE client32g socket259 210 5443-5464
START_RANGE client32h socket260 220 5464-5486
START_RANGE client32i socket261 210 5486-5507
START_RANGE client32j socket262 210 5507-5528
START_RANGE client32k socket263 210 5528-5549
START_RANGE client32l socket264 210 5549-5570
START_RANGE client32a socket265 210 5570-5591
START_RANGE client32b socket266 210 5591-5612
START_RANGE client32c socket267 210 5612-5633
START_RANGE client32d socket268 210 5633-5654
START_RANGE client32e socket269 210 5654-5675
START_RANGE client32f socket270 220 5675-5697
START_RANGE client32g socket271 210 5697-5718
START_RANGE client32h socket272 210 5718-5739
START_RANGE client32i socket273 210 5739-5760
START_RANGE client32j socket274 210 5760-5781
START_RANGE client32k socket275 210 5781-5802
START_RANGE client32l socket276 210 5802-5823
START_RANGE client32a socket277 210 5823-5844
START_RANGE client32b socket278 210 5844-5865
START_RANGE client32c socket279 210 5865-5886
START_RANGE client32d socket280 220 5886-5908
START_RANGE client32e socket281 210 5908-5929
START_RANGE client32f socket282 210 5929-5950
START_RANGE client32g socket283 210 5950-5971
START_RANGE client32h socket284 210 5971-5992
START_RANGE client32i socket285 210 5992-6013
START_RANGE client32j socket286 210 6013-6034
START_RANGE client32k socket287 210 6034-6055
START_RANGE client32l socket288 210 6055-6076

#elif MASTER_NUM2
START_RANGE client41a socket289 210 6076-6097
START_RANGE client41b socket290 220 6097-6119
START_RANGE client41c socket291 210 6119-6140
START_RANGE client41d socket292 210 6140-6161
START_RANGE client41e socket293 210 6161-6182
START_RANGE client41f socket294 210 6182-6203
START_RANGE client41g socket295 210 6203-6224
START_RANGE client41h socket296 210 6224-6245
START_RANGE client41i socket297 210 6245-6266
START_RANGE client41j socket298 210 6266-6287
START_RANGE client41k socket299 210 6287-6308
START_RANGE client41l socket300 220 6308-6330
START_RANGE client41a socket301 210 6330-6351
START_RANGE client41b socket302 210 6351-6372
START_RANGE client41c socket303 210 6372-6393
START_RANGE client41d socket304 210 6393-6414
START_RANGE client41e socket305 210 6414-6435
START_RANGE client41f socket306 210 6435-6456
START_RANGE client41g socket307 210 6456-6477
START_RANGE client41h socket308 210 6477-6498
START_RANGE client41i socket309 210 6498-6519
START_RANGE client41j socket310 220 6519-6541
START_RANGE client41k socket311 210 6541-6562
START_RANGE client41l socket312 210 6562-6583
START_RANGE client41a socket313 210 6583-6604
START_RANGE client41b socket314 210 6604-6625
```

START_RANGE client41c socket315 210 6625-6646
START_RANGE client41d socket316 210 6646-6667
START_RANGE client41e socket317 210 6667-6688
START_RANGE client41f socket318 210 6688-6709
START_RANGE client41g socket319 210 6709-6730
START_RANGE client41h socket320 220 6730-6752
START_RANGE client41i socket321 210 6752-6773
START_RANGE client41j socket322 210 6773-6794
START_RANGE client41k socket323 210 6794-6815
START_RANGE client41l socket324 210 6815-6836
START_RANGE client41a socket325 210 6836-6857
START_RANGE client41b socket326 210 6857-6878
START_RANGE client41c socket327 210 6878-6899
START_RANGE client41d socket328 210 6899-6920
START_RANGE client41e socket329 210 6920-6941
START_RANGE client41f socket330 220 6941-6963
START_RANGE client41g socket331 210 6963-6984
START_RANGE client41h socket332 210 6984-7005
START_RANGE client41i socket333 210 7005-7026
START_RANGE client41j socket334 210 7026-7047
START_RANGE client41k socket335 210 7047-7068
START_RANGE client41l socket336 210 7068-7089

START_RANGE client42a socket337 210 7089-7110
START_RANGE client42b socket338 210 7110-7131
START_RANGE client42c socket339 210 7131-7152
START_RANGE client42d socket340 220 7152-7174
START_RANGE client42e socket341 210 7174-7195
START_RANGE client42f socket342 210 7195-7216
START_RANGE client42g socket343 210 7216-7237
START_RANGE client42h socket344 210 7237-7258
START_RANGE client42i socket345 210 7258-7279
START_RANGE client42j socket346 210 7279-7300
START_RANGE client42k socket347 210 7300-7321
START_RANGE client42l socket348 210 7321-7342
START_RANGE client42a socket349 210 7342-7363
START_RANGE client42b socket350 220 7363-7385
START_RANGE client42c socket351 210 7385-7406
START_RANGE client42d socket352 210 7406-7427
START_RANGE client42e socket353 210 7427-7448
START_RANGE client42f socket354 210 7448-7469
START_RANGE client42g socket355 210 7469-7490
START_RANGE client42h socket356 210 7490-7511
START_RANGE client42i socket357 210 7511-7532
START_RANGE client42j socket358 210 7532-7553
START_RANGE client42k socket359 210 7553-7574
START_RANGE client42l socket360 220 7574-7596
START_RANGE client42a socket361 210 7596-7617
START_RANGE client42b socket362 210 7617-7638
START_RANGE client42c socket363 210 7638-7659
START_RANGE client42d socket364 210 7659-7680
START_RANGE client42e socket365 210 7680-7701
START_RANGE client42f socket366 210 7701-7722
START_RANGE client42g socket367 210 7722-7743
START_RANGE client42h socket368 210 7743-7764
START_RANGE client42i socket369 210 7764-7785
START_RANGE client42j socket370 220 7785-7807
START_RANGE client42k socket371 210 7807-7828
START_RANGE client42l socket372 210 7828-7849
START_RANGE client42a socket373 210 7849-7870
START_RANGE client42b socket374 210 7870-7891
START_RANGE client42c socket375 210 7891-7912
START_RANGE client42d socket376 210 7912-7933
START_RANGE client42e socket377 210 7933-7954
START_RANGE client42f socket378 210 7954-7975
START_RANGE client42g socket379 210 7975-7996
START_RANGE client42h socket380 220 7996-8018
START_RANGE client42i socket381 210 8018-8039
START_RANGE client42j socket382 210 8039-8060
START_RANGE client42k socket383 210 8060-8081
START_RANGE client42l socket384 210 8081-8102

START_RANGE client51a socket385 210 8102-8123
START_RANGE client51b socket386 210 8123-8144
START_RANGE client51c socket387 210 8144-8165
START_RANGE client51d socket388 210 8165-8186
START_RANGE client51e socket389 210 8186-8207
START_RANGE client51f socket390 220 8207-8229
START_RANGE client51g socket391 210 8229-8250
START_RANGE client51h socket392 210 8250-8271
START_RANGE client51i socket393 210 8271-8292
START_RANGE client51j socket394 210 8292-8313
START_RANGE client51k socket395 210 8313-8334
START_RANGE client51l socket396 210 8334-8355
START_RANGE client51a socket397 210 8355-8376
START_RANGE client51b socket398 210 8376-8397
START_RANGE client51c socket399 210 8397-8418
START_RANGE client51d socket400 220 8418-8440
START_RANGE client51e socket401 210 8440-8461
START_RANGE client51f socket402 210 8461-8482
START_RANGE client51g socket403 210 8482-8503
START_RANGE client51h socket404 210 8503-8524
START_RANGE client51i socket405 210 8524-8545
START_RANGE client51j socket406 210 8545-8566
START_RANGE client51k socket407 210 8566-8587
START_RANGE client51l socket408 210 8587-8608
START_RANGE client51a socket409 210 8608-8629
START_RANGE client51b socket410 220 8629-8651
START_RANGE client51c socket411 210 8651-8672
START_RANGE client51d socket412 210 8672-8693
START_RANGE client51e socket413 210 8693-8714
START_RANGE client51f socket414 210 8714-8735
START_RANGE client51g socket415 210 8735-8756
START_RANGE client51h socket416 210 8756-8777
START_RANGE client51i socket417 210 8777-8798
START_RANGE client51j socket418 210 8798-8819
START_RANGE client51k socket419 210 8819-8840
START_RANGE client51l socket420 220 8840-8862
START_RANGE client51a socket421 210 8862-8883
START_RANGE client51b socket422 210 8883-8904
START_RANGE client51c socket423 210 8904-8925
START_RANGE client51d socket424 210 8925-8946
START_RANGE client51e socket425 210 8946-8967
START_RANGE client51f socket426 210 8967-8988
START_RANGE client51g socket427 210 8988-9009
START_RANGE client51h socket428 210 9009-9030
START_RANGE client51i socket429 210 9030-9051
START_RANGE client51j socket430 220 9051-9073
START_RANGE client51k socket431 210 9073-9094
START_RANGE client51l socket432 210 9094-9115

START_RANGE client52a socket433 210 9115-9136
START_RANGE client52b socket434 210 9136-9157
START_RANGE client52c socket435 210 9157-9178
START_RANGE client52d socket436 210 9178-9199
START_RANGE client52e socket437 210 9199-9220
START_RANGE client52f socket438 210 9220-9241
START_RANGE client52g socket439 210 9241-9262
START_RANGE client52h socket440 220 9262-9284
START_RANGE client52i socket441 210 9284-9305
START_RANGE client52j socket442 210 9305-9326
START_RANGE client52k socket443 210 9326-9347
START_RANGE client52l socket444 210 9347-9368
START_RANGE client52a socket445 210 9368-9389
START_RANGE client52b socket446 210 9389-9410
START_RANGE client52c socket447 210 9410-9431
START_RANGE client52d socket448 210 9431-9452
START_RANGE client52e socket449 210 9452-9473
START_RANGE client52f socket450 220 9473-9495
START_RANGE client52g socket451 210 9495-9516
START_RANGE client52h socket452 210 9516-9537
START_RANGE client52i socket453 210 9537-9558
START_RANGE client52j socket454 210 9558-9579
START_RANGE client52k socket455 210 9579-9600
START_RANGE client52l socket456 210 9600-9621
START_RANGE client52a socket457 210 9621-9642
START_RANGE client52b socket458 210 9642-9663
START_RANGE client52c socket459 210 9663-9684
START_RANGE client52d socket460 220 9684-9706
START_RANGE client52e socket461 210 9706-9727
START_RANGE client52f socket462 210 9727-9748
START_RANGE client52g socket463 210 9748-9769
START_RANGE client52h socket464 210 9769-9790
START_RANGE client52i socket465 210 9790-9811
START_RANGE client52j socket466 210 9811-9832
START_RANGE client52k socket467 210 9832-9853
START_RANGE client52l socket468 210 9853-9874
START_RANGE client52a socket469 210 9874-9895
START_RANGE client52b socket470 220 9895-9917
START_RANGE client52c socket471 210 9917-9938
START_RANGE client52d socket472 210 9938-9959
START_RANGE client52e socket473 210 9959-9980
START_RANGE client52f socket474 210 9980-10001
START_RANGE client52g socket475 210 10001-10022
START_RANGE client52h socket476 210 10022-10043
START_RANGE client52i socket477 210 10043-10064
START_RANGE client52j socket478 210 10064-10085
START_RANGE client52k socket479 210 10085-10106
START_RANGE client52l socket480 220 10106-10128

START_RANGE client61a socket481 210 10128-10149
START_RANGE client61b socket482 210 10149-10170
START_RANGE client61c socket483 210 10170-10191
START_RANGE client61d socket484 210 10191-10212
START_RANGE client61e socket485 210 10212-10233
START_RANGE client61f socket486 210 10233-10254
START_RANGE client61g socket487 210 10254-10275
START_RANGE client61h socket488 210 10275-10296
START_RANGE client61i socket489 210 10296-10317
START_RANGE client61j socket490 220 10317-10339
START_RANGE client61k socket491 210 10339-10360
START_RANGE client61l socket492 210 10360-10381
START_RANGE client61a socket493 210 10381-10402
START_RANGE client61b socket494 210 10402-10423
START_RANGE client61c socket495 210 10423-10444
START_RANGE client61d socket496 210 10444-10465
START_RANGE client61e socket497 210 10465-10486
START_RANGE client61f socket498 210 10486-10507
START_RANGE client61g socket499 210 10507-10528
START_RANGE client61h socket500 220 10528-10550
START_RANGE client61i socket501 210 10550-10571
START_RANGE client61j socket502 210 10571-10592
START_RANGE client61k socket503 210 10592-10613
START_RANGE client61l socket504 210 10613-10634
START_RANGE client61a socket505 210 10634-10655
START_RANGE client61b socket506 210 10655-10676
START_RANGE client61c socket507 210 10676-10697
START_RANGE client61d socket508 210 10697-10718

START_RANGE client61e socket509 210 10718-10739
START_RANGE client61f socket510 220 10739-10761
START_RANGE client61g socket511 210 10761-10782
START_RANGE client61h socket512 210 10782-10803
START_RANGE client61i socket513 210 10803-10824
START_RANGE client61j socket514 210 10824-10845
START_RANGE client61k socket515 210 10845-10866
START_RANGE client61l socket516 210 10866-10887
START_RANGE client61a socket517 210 10887-10908
START_RANGE client61b socket518 210 10908-10929
START_RANGE client61c socket519 210 10929-10950
START_RANGE client61d socket520 220 10950-10972
START_RANGE client61e socket521 210 10972-10993
START_RANGE client61f socket522 210 10993-11014
START_RANGE client61g socket523 210 11014-11035
START_RANGE client61h socket524 210 11035-11056
START_RANGE client61i socket525 210 11056-11077
START_RANGE client61j socket526 210 11077-11098
START_RANGE client61k socket527 210 11098-11119
START_RANGE client61l socket528 210 11119-11140

START_RANGE client62a socket529 210 11140-11161
START_RANGE client62b socket530 220 11161-11183
START_RANGE client62c socket531 210 11183-11204
START_RANGE client62d socket532 210 11204-11225
START_RANGE client62e socket533 210 11225-11246
START_RANGE client62f socket534 210 11246-11267
START_RANGE client62g socket535 210 11267-11288
START_RANGE client62h socket536 210 11288-11309
START_RANGE client62i socket537 210 11309-11330
START_RANGE client62j socket538 210 11330-11351
START_RANGE client62k socket539 210 11351-11372
START_RANGE client62l socket540 220 11372-11394
START_RANGE client62a socket541 210 11394-11415
START_RANGE client62b socket542 210 11415-11436
START_RANGE client62c socket543 210 11436-11457
START_RANGE client62d socket544 210 11457-11478
START_RANGE client62e socket545 210 11478-11499
START_RANGE client62f socket546 210 11499-11520
START_RANGE client62g socket547 210 11520-11541
START_RANGE client62h socket548 210 11541-11562
START_RANGE client62i socket549 210 11562-11583
START_RANGE client62j socket550 220 11583-11605
START_RANGE client62k socket551 210 11605-11626
START_RANGE client62l socket552 210 11626-11647
START_RANGE client62a socket553 210 11647-11668
START_RANGE client62b socket554 210 11668-11689
START_RANGE client62c socket555 210 11689-11710
START_RANGE client62d socket556 210 11710-11731
START_RANGE client62e socket557 210 11731-11752
START_RANGE client62f socket558 210 11752-11773
START_RANGE client62g socket559 210 11773-11794
START_RANGE client62h socket560 220 11794-11816
START_RANGE client62i socket561 210 11816-11837
START_RANGE client62j socket562 210 11837-11858
START_RANGE client62k socket563 210 11858-11879
START_RANGE client62l socket564 210 11879-11900
START_RANGE client62a socket565 210 11900-11921
START_RANGE client62b socket566 210 11921-11942
START_RANGE client62c socket567 210 11942-11963
START_RANGE client62d socket568 210 11963-11984
START_RANGE client62e socket569 210 11984-12005
START_RANGE client62f socket570 220 12005-12027
START_RANGE client62g socket571 210 12027-12048
START_RANGE client62h socket572 210 12048-12069
START_RANGE client62i socket573 210 12069-12090
START_RANGE client62j socket574 210 12090-12111
START_RANGE client62k socket575 210 12111-12132
START_RANGE client62l socket576 210 12132-12153

#elif MASTER_NUM3
START_RANGE client71a socket577 210 12153-12174
START_RANGE client71b socket578 210 12174-12195
START_RANGE client71c socket579 210 12195-12216
START_RANGE client71d socket580 220 12216-12238
START_RANGE client71e socket581 210 12238-12259
START_RANGE client71f socket582 210 12259-12280
START_RANGE client71g socket583 210 12280-12301
START_RANGE client71h socket584 210 12301-12322
START_RANGE client71i socket585 210 12322-12343
START_RANGE client71j socket586 210 12343-12364
START_RANGE client71k socket587 210 12364-12385
START_RANGE client71l socket588 210 12385-12406
START_RANGE client71a socket589 210 12406-12427
START_RANGE client71b socket590 220 12427-12449
START_RANGE client71c socket591 210 12449-12470
START_RANGE client71d socket592 210 12470-12491
START_RANGE client71e socket593 210 12491-12512
START_RANGE client71f socket594 210 12512-12533
START_RANGE client71g socket595 210 12533-12554
START_RANGE client71h socket596 210 12554-12575
START_RANGE client71i socket597 210 12575-12596
START_RANGE client71j socket598 210 12596-12617
START_RANGE client71k socket599 210 12617-12638
START_RANGE client71l socket600 220 12638-12660
START_RANGE client71a socket601 210 12660-12681
START_RANGE client71b socket602 210 12681-12702
START_RANGE client71c socket603 210 12702-12723
START_RANGE client71d socket604 210 12723-12744
START_RANGE client71e socket605 210 12744-12765
START_RANGE client71f socket606 210 12765-12786
START_RANGE client71g socket607 210 12786-12807
START_RANGE client71h socket608 210 12807-12828
START_RANGE client71i socket609 210 12828-12849
START_RANGE client71j socket610 220 12849-12871
START_RANGE client71k socket611 210 12871-12892
START_RANGE client71l socket612 210 12892-12913
START_RANGE client71a socket613 210 12913-12934
START_RANGE client71b socket614 210 12934-12955
START_RANGE client71c socket615 210 12955-12976
START_RANGE client71d socket616 210 12976-12997
START_RANGE client71e socket617 210 12997-13018
START_RANGE client71f socket618 210 13018-13039
START_RANGE client71g socket619 210 13039-13060
START_RANGE client71h socket620 220 13060-13082
START_RANGE client71i socket621 210 13082-13103
START_RANGE client71j socket622 210 13103-13124
START_RANGE client71k socket623 210 13124-13145
START_RANGE client71l socket624 210 13145-13166

START_RANGE client72a socket625 210 13166-13187
START_RANGE client72b socket626 210 13187-13208
START_RANGE client72c socket627 210 13208-13229
START_RANGE client72d socket628 210 13229-13250
START_RANGE client72e socket629 210 13250-13271
START_RANGE client72f socket630 220 13271-13293
START_RANGE client72g socket631 210 13293-13314
START_RANGE client72h socket632 210 13314-13335
START_RANGE client72i socket633 210 13335-13356
START_RANGE client72j socket634 210 13356-13377
START_RANGE client72k socket635 210 13377-13398
START_RANGE client72l socket636 210 13398-13419
START_RANGE client72a socket637 210 13419-13440
START_RANGE client72b socket638 210 13440-13461
START_RANGE client72c socket639 210 13461-13482
START_RANGE client72d socket640 220 13482-13504
START_RANGE client72e socket641 210 13504-13525
START_RANGE client72f socket642 210 13525-13546
START_RANGE client72g socket643 210 13546-13567
START_RANGE client72h socket644 210 13567-13588
START_RANGE client72i socket645 210 13588-13609
START_RANGE client72j socket646 210 13609-13630
START_RANGE client72k socket647 210 13630-13651
START_RANGE client72l socket648 210 13651-13672
START_RANGE client72a socket649 210 13672-13693
START_RANGE client72b socket650 220 13693-13715
START_RANGE client72c socket651 210 13715-13736
START_RANGE client72d socket652 210 13736-13757
START_RANGE client72e socket653 210 13757-13778
START_RANGE client72f socket654 210 13778-13799
START_RANGE client72g socket655 210 13799-13820
START_RANGE client72h socket656 210 13820-13841
START_RANGE client72i socket657 210 13841-13862
START_RANGE client72j socket658 210 13862-13883
START_RANGE client72k socket659 210 13883-13904
START_RANGE client72l socket660 220 13904-13926
START_RANGE client72a socket661 210 13926-13947
START_RANGE client72b socket662 210 13947-13968
START_RANGE client72c socket663 210 13968-13989
START_RANGE client72d socket664 210 13989-14010
START_RANGE client72e socket665 210 14010-14031
START_RANGE client72f socket666 210 14031-14052
START_RANGE client72g socket667 210 14052-14073
START_RANGE client72h socket668 210 14073-14094
START_RANGE client72i socket669 210 14094-14115
START_RANGE client72j socket670 220 14115-14137
START_RANGE client72k socket671 210 14137-14158
START_RANGE client72l socket672 210 14158-14179

START_RANGE client81a socket673 210 14179-14200
START_RANGE client81b socket674 210 14200-14221
START_RANGE client81c socket675 210 14221-14242
START_RANGE client81d socket676 210 14242-14263
START_RANGE client81e socket677 210 14263-14284
START_RANGE client81f socket678 210 14284-14305
START_RANGE client81g socket679 210 14305-14326
START_RANGE client81h socket680 220 14326-14348
START_RANGE client81i socket681 210 14348-14369
START_RANGE client81j socket682 210 14369-14390
START_RANGE client81k socket683 210 14390-14411
START_RANGE client81l socket684 210 14411-14432
START_RANGE client81a socket685 210 14432-14453
START_RANGE client81b socket686 210 14453-14474
START_RANGE client81c socket687 210 14474-14495
START_RANGE client81d socket688 210 14495-14516
START_RANGE client81e socket689 210 14516-14537
START_RANGE client81f socket690 220 14537-14559
START_RANGE client81g socket691 210 14559-14580
START_RANGE client81h socket692 210 14580-14601
START_RANGE client81i socket693 210 14601-14622
START_RANGE client81j socket694 210 14622-14643
START_RANGE client81k socket695 210 14643-14664
START_RANGE client81l socket696 210 14664-14685
START_RANGE client81a socket697 210 14685-14706
START_RANGE client81b socket698 210 14706-14727
START_RANGE client81c socket699 210 14727-14748
START_RANGE client81d socket700 220 14748-14770
START_RANGE client81e socket701 210 14770-14791

START_RANGE client81f socket702 210 14791-14812
START_RANGE client81g socket703 210 14812-14833
START_RANGE client81h socket704 210 14833-14854
START_RANGE client81i socket705 210 14854-14875
START_RANGE client81j socket706 210 14875-14896
START_RANGE client81k socket707 210 14896-14917
START_RANGE client81l socket708 210 14917-14938
START_RANGE client81a socket709 210 14938-14959
START_RANGE client81b socket710 220 14959-14981
START_RANGE client81c socket711 210 14981-15002
START_RANGE client81d socket712 210 15002-15023
START_RANGE client81e socket713 210 15023-15044
START_RANGE client81f socket714 210 15044-15065
START_RANGE client81g socket715 210 15065-15086
START_RANGE client81h socket716 210 15086-15107
START_RANGE client81i socket717 210 15107-15128
START_RANGE client81j socket718 210 15128-15149
START_RANGE client81k socket719 210 15149-15170
START_RANGE client81l socket720 220 15170-15192

START_RANGE client82a socket721 210 15192-15213
START_RANGE client82b socket722 210 15213-15234
START_RANGE client82c socket723 210 15234-15255
START_RANGE client82d socket724 210 15255-15276
START_RANGE client82e socket725 210 15276-15297
START_RANGE client82f socket726 210 15297-15318
START_RANGE client82g socket727 210 15318-15339
START_RANGE client82h socket728 210 15339-15360
START_RANGE client82i socket729 210 15360-15381
START_RANGE client82j socket730 220 15381-15403
START_RANGE client82k socket731 210 15403-15424
START_RANGE client82l socket732 210 15424-15445
START_RANGE client82a socket733 210 15445-15466
START_RANGE client82b socket734 210 15466-15487
START_RANGE client82c socket735 210 15487-15508
START_RANGE client82d socket736 210 15508-15529
START_RANGE client82e socket737 210 15529-15550
START_RANGE client82f socket738 210 15550-15571
START_RANGE client82g socket739 210 15571-15592
START_RANGE client82h socket740 220 15592-15614
START_RANGE client82i socket741 210 15614-15635
START_RANGE client82j socket742 210 15635-15656
START_RANGE client82k socket743 210 15656-15677
START_RANGE client82l socket744 210 15677-15698
START_RANGE client82a socket745 210 15698-15719
START_RANGE client82b socket746 210 15719-15740
START_RANGE client82c socket747 210 15740-15761
START_RANGE client82d socket748 210 15761-15782
START_RANGE client82e socket749 210 15782-15803
START_RANGE client82f socket750 220 15803-15825
START_RANGE client82g socket751 210 15825-15846
START_RANGE client82h socket752 210 15846-15867
START_RANGE client82i socket753 210 15867-15888
START_RANGE client82j socket754 210 15888-15909
START_RANGE client82k socket755 210 15909-15930
START_RANGE client82l socket756 210 15930-15951
START_RANGE client82a socket757 210 15951-15972
START_RANGE client82b socket758 210 15972-15993
START_RANGE client82c socket759 210 15993-16014
START_RANGE client82d socket760 220 16014-16036
START_RANGE client82e socket761 210 16036-16057
START_RANGE client82f socket762 210 16057-16078
START_RANGE client82g socket763 210 16078-16099
START_RANGE client82h socket764 210 16099-16120
START_RANGE client82i socket765 210 16120-16141
START_RANGE client82j socket766 210 16141-16162
START_RANGE client82k socket767 210 16162-16183
START_RANGE client82l socket768 210 16183-16204

START_RANGE client91a socket769 210 16204-16225
START_RANGE client91b socket770 220 16225-16247
START_RANGE client91c socket771 210 16247-16268
START_RANGE client91d socket772 210 16268-16289
START_RANGE client91e socket773 210 16289-16310
START_RANGE client91f socket774 210 16310-16331
START_RANGE client91g socket775 210 16331-16352
START_RANGE client91h socket776 210 16352-16373
START_RANGE client91i socket777 210 16373-16394
START_RANGE client91j socket778 210 16394-16415
START_RANGE client91k socket779 210 16415-16436
START_RANGE client91l socket780 220 16436-16458
START_RANGE client91a socket781 210 16458-16479
START_RANGE client91b socket782 210 16479-16500
START_RANGE client91c socket783 210 16500-16521
START_RANGE client91d socket784 210 16521-16542
START_RANGE client91e socket785 210 16542-16563
START_RANGE client91f socket786 210 16563-16584
START_RANGE client91g socket787 210 16584-16605
START_RANGE client91h socket788 210 16605-16626
START_RANGE client91i socket789 210 16626-16647
START_RANGE client91j socket790 220 16647-16669
START_RANGE client91k socket791 210 16669-16690
START_RANGE client91l socket792 210 16690-16711
START_RANGE client91a socket793 210 16711-16732
START_RANGE client91b socket794 210 16732-16753
START_RANGE client91c socket795 210 16753-16774
START_RANGE client91d socket796 210 16774-16795
START_RANGE client91e socket797 210 16795-16816
START_RANGE client91f socket798 210 16816-16837
START_RANGE client91g socket799 210 16837-16858
START_RANGE client91h socket800 220 16858-16880
START_RANGE client91i socket801 210 16880-16901
START_RANGE client91j socket802 210 16901-16922
START_RANGE client91k socket803 210 16922-16943
START_RANGE client91l socket804 210 16943-16964
START_RANGE client91a socket805 210 16964-16985
START_RANGE client91b socket806 210 16985-17006
START_RANGE client91c socket807 210 17006-17027
START_RANGE client91d socket808 210 17027-17048
START_RANGE client91e socket809 210 17048-17069
START_RANGE client91f socket810 220 17069-17091
START_RANGE client91g socket811 210 17091-17112
START_RANGE client91h socket812 210 17112-17133
START_RANGE client91i socket813 210 17133-17154
START_RANGE client91j socket814 210 17154-17175
START_RANGE client91k socket815 210 17175-17196
START_RANGE client91l socket816 210 17196-17217

START_RANGE client92a socket817 210 17217-17238
START_RANGE client92b socket818 210 17238-17259
START_RANGE client92c socket819 210 17259-17280
START_RANGE client92d socket820 220 17280-17302
START_RANGE client92e socket821 210 17302-17323
START_RANGE client92f socket822 210 17323-17344
START_RANGE client92g socket823 210 17344-17365
START_RANGE client92h socket824 210 17365-17386
START_RANGE client92i socket825 210 17386-17407
START_RANGE client92j socket826 210 17407-17428
START_RANGE client92k socket827 210 17428-17449
START_RANGE client92l socket828 210 17449-17470
START_RANGE client92a socket829 210 17470-17491
START_RANGE client92b socket830 220 17491-17513
START_RANGE client92c socket831 210 17513-17534
START_RANGE client92d socket832 210 17534-17555
START_RANGE client92e socket833 210 17555-17576
START_RANGE client92f socket834 210 17576-17597
START_RANGE client92g socket835 210 17597-17618
START_RANGE client92h socket836 210 17618-17639
START_RANGE client92i socket837 210 17639-17660
START_RANGE client92j socket838 210 17660-17681
START_RANGE client92k socket839 210 17681-17702
START_RANGE client92l socket840 220 17702-17724
START_RANGE client92a socket841 210 17724-17745
START_RANGE client92b socket842 210 17745-17766
START_RANGE client92c socket843 210 17766-17787
START_RANGE client92d socket844 210 17787-17808
START_RANGE client92e socket845 210 17808-17829
START_RANGE client92f socket846 210 17829-17850
START_RANGE client92g socket847 210 17850-17871
START_RANGE client92h socket848 210 17871-17892
START_RANGE client92i socket849 210 17892-17913
START_RANGE client92j socket850 220 17913-17935
START_RANGE client92k socket851 210 17935-17956
START_RANGE client92l socket852 210 17956-17977
START_RANGE client92a socket853 210 17977-17998
START_RANGE client92b socket854 210 17998-18019
START_RANGE client92c socket855 210 18019-18040
START_RANGE client92d socket856 210 18040-18061
START_RANGE client92e socket857 210 18061-18082
START_RANGE client92f socket858 210 18082-18103
START_RANGE client92g socket859 210 18103-18124
START_RANGE client92h socket860 220 18124-18146
START_RANGE client92i socket861 210 18146-18167
START_RANGE client92j socket862 210 18167-18188
START_RANGE client92k socket863 210 18188-18209
START_RANGE client92l socket864 210 18209-18230

#elif MASTER_NUM4
START_RANGE client101a socket865 210 18230-18251
START_RANGE client101b socket866 210 18251-18272
START_RANGE client101c socket867 210 18272-18293
START_RANGE client101d socket868 210 18293-18314
START_RANGE client101e socket869 210 18314-18335
START_RANGE client101f socket870 220 18335-18357
START_RANGE client101g socket871 210 18357-18378
START_RANGE client101h socket872 210 18378-18399
START_RANGE client101i socket873 210 18399-18420
START_RANGE client101j socket874 210 18420-18441
START_RANGE client101k socket875 210 18441-18462
START_RANGE client101l socket876 210 18462-18483
START_RANGE client101a socket877 210 18483-18504
START_RANGE client101b socket878 210 18504-18525
START_RANGE client101c socket879 210 18525-18546
START_RANGE client101d socket880 220 18546-18568
START_RANGE client101e socket881 210 18568-18589
START_RANGE client101f socket882 210 18589-18610
START_RANGE client101g socket883 210 18610-18631
START_RANGE client101h socket884 210 18631-18652
START_RANGE client101i socket885 210 18652-18673
START_RANGE client101j socket886 210 18673-18694
START_RANGE client101k socket887 210 18694-18715
START_RANGE client101l socket888 210 18715-18736
START_RANGE client101a socket889 210 18736-18757
START_RANGE client101b socket890 220 18757-18779
START_RANGE client101c socket891 210 18779-18800
START_RANGE client101d socket892 210 18800-18821
START_RANGE client101e socket893 210 18821-18842
START_RANGE client101f socket894 210 18842-18863

START_RANGE client101g socket895 210 18863-18884
START_RANGE client101h socket896 210 18884-18905
START_RANGE client101i socket897 210 18905-18926
START_RANGE client101j socket898 210 18926-18947
START_RANGE client101k socket899 210 18947-18968
START_RANGE client101l socket900 220 18968-18990
START_RANGE client101a socket901 210 18990-19011
START_RANGE client101b socket902 210 19011-19032
START_RANGE client101c socket903 210 19032-19053
START_RANGE client101d socket904 210 19053-19074
START_RANGE client101e socket905 210 19074-19095
START_RANGE client101f socket906 210 19095-19116
START_RANGE client101g socket907 210 19116-19137
START_RANGE client101h socket908 210 19137-19158
START_RANGE client101i socket909 210 19158-19179
START_RANGE client101j socket910 220 19179-19201
START_RANGE client101k socket911 210 19201-19222
START_RANGE client101l socket912 210 19222-19243

START_RANGE client102a socket913 210 19243-19264
START_RANGE client102b socket914 210 19264-19285
START_RANGE client102c socket915 210 19285-19306
START_RANGE client102d socket916 210 19306-19327
START_RANGE client102e socket917 210 19327-19348
START_RANGE client102f socket918 210 19348-19369
START_RANGE client102g socket919 210 19369-19390
START_RANGE client102h socket920 220 19390-19412
START_RANGE client102i socket921 210 19412-19433
START_RANGE client102j socket922 210 19433-19454
START_RANGE client102k socket923 210 19454-19475
START_RANGE client102l socket924 210 19475-19496
START_RANGE client102a socket925 210 19496-19517
START_RANGE client102b socket926 210 19517-19538
START_RANGE client102c socket927 210 19538-19559
START_RANGE client102d socket928 210 19559-19580
START_RANGE client102e socket929 210 19580-19601
START_RANGE client102f socket930 220 19601-19623
START_RANGE client102g socket931 210 19623-19644
START_RANGE client102h socket932 210 19644-19665
START_RANGE client102i socket933 210 19665-19686
START_RANGE client102j socket934 210 19686-19707
START_RANGE client102k socket935 210 19707-19728
START_RANGE client102l socket936 210 19728-19749
START_RANGE client102a socket937 210 19749-19770
START_RANGE client102b socket938 210 19770-19791
START_RANGE client102c socket939 210 19791-19812
START_RANGE client102d socket940 220 19812-19834
START_RANGE client102e socket941 210 19834-19855
START_RANGE client102f socket942 210 19855-19876
START_RANGE client102g socket943 210 19876-19897
START_RANGE client102h socket944 210 19897-19918
START_RANGE client102i socket945 210 19918-19939
START_RANGE client102j socket946 210 19939-19960
START_RANGE client102k socket947 210 19960-19981
START_RANGE client102l socket948 210 19981-20002
START_RANGE client102a socket949 210 20002-20023
START_RANGE client102b socket950 220 20023-20045
START_RANGE client102c socket951 210 20045-20066
START_RANGE client102d socket952 210 20066-20087
START_RANGE client102e socket953 210 20087-20108
START_RANGE client102f socket954 210 20108-20129
START_RANGE client102g socket955 210 20129-20150
START_RANGE client102h socket956 210 20150-20171
START_RANGE client102i socket957 210 20171-20192
START_RANGE client102j socket958 210 20192-20213
START_RANGE client102k socket959 210 20213-20234
START_RANGE client102l socket960 220 20234-20256

START_RANGE client111a socket961 210 20256-20277
START_RANGE client111b socket962 210 20277-20298
START_RANGE client111c socket963 210 20298-20319
START_RANGE client111d socket964 210 20319-20340
START_RANGE client111e socket965 210 20340-20361
START_RANGE client111f socket966 210 20361-20382
START_RANGE client111g socket967 210 20382-20403
START_RANGE client111h socket968 210 20403-20424
START_RANGE client111i socket969 210 20424-20445
START_RANGE client111j socket970 220 20445-20467
START_RANGE client111k socket971 210 20467-20488
START_RANGE client111l socket972 210 20488-20509
START_RANGE client111a socket973 210 20509-20530
START_RANGE client111b socket974 210 20530-20551
START_RANGE client111c socket975 210 20551-20572
START_RANGE client111d socket976 210 20572-20593
START_RANGE client111e socket977 210 20593-20614
START_RANGE client111f socket978 210 20614-20635
START_RANGE client111g socket979 210 20635-20656
START_RANGE client111h socket980 220 20656-20678
START_RANGE client111i socket981 210 20678-20699
START_RANGE client111j socket982 210 20699-20720
START_RANGE client111k socket983 210 20720-20741
START_RANGE client111l socket984 210 20741-20762
START_RANGE client111a socket985 210 20762-20783
START_RANGE client111b socket986 210 20783-20804
START_RANGE client111c socket987 210 20804-20825
START_RANGE client111d socket988 210 20825-20846
START_RANGE client111e socket989 210 20846-20867
START_RANGE client111f socket990 220 20867-20889
START_RANGE client111g socket991 210 20889-20910
START_RANGE client111h socket992 210 20910-20931
START_RANGE client111i socket993 210 20931-20952
START_RANGE client111j socket994 210 20952-20973
START_RANGE client111k socket995 210 20973-20994
START_RANGE client111l socket996 210 20994-21015
START_RANGE client111a socket997 210 21015-21036
START_RANGE client111b socket998 210 21036-21057
START_RANGE client111c socket999 210 21057-21078
START_RANGE client111d socket1000 220 21078-21100
START_RANGE client111e socket1001 210 21100-21121
START_RANGE client111f socket1002 210 21121-21142
START_RANGE client111g socket1003 210 21142-21163
START_RANGE client111h socket1004 210 21163-21184
START_RANGE client111i socket1005 210 21184-21205
START_RANGE client111j socket1006 210 21205-21226
START_RANGE client111k socket1007 210 21226-21247
START_RANGE client111l socket1008 210 21247-21268

START_RANGE client112a socket1009 210 21268-21289
START_RANGE client112b socket1010 220 21289-21311
START_RANGE client112c socket1011 210 21311-21332
START_RANGE client112d socket1012 210 21332-21353
START_RANGE client112e socket1013 210 21353-21374
START_RANGE client112f socket1014 210 21374-21395
START_RANGE client112g socket1015 210 21395-21416
START_RANGE client112h socket1016 210 21416-21437
START_RANGE client112i socket1017 210 21437-21458
START_RANGE client112j socket1018 210 21458-21479
START_RANGE client112k socket1019 210 21479-21500
START_RANGE client112l socket1020 220 21500-21522
START_RANGE client112a socket1021 210 21522-21543
START_RANGE client112b socket1022 210 21543-21564
START_RANGE client112c socket1023 210 21564-21585
START_RANGE client112d socket1024 210 21585-21606
START_RANGE client112e socket1025 210 21606-21627
START_RANGE client112f socket1026 210 21627-21648
START_RANGE client112g socket1027 210 21648-21669
START_RANGE client112h socket1028 210 21669-21690
START_RANGE client112i socket1029 210 21690-21711
START_RANGE client112j socket1030 220 21711-21733
START_RANGE client112k socket1031 210 21733-21754
START_RANGE client112l socket1032 210 21754-21775
START_RANGE client112a socket1033 210 21775-21796
START_RANGE client112b socket1034 210 21796-21817
START_RANGE client112c socket1035 210 21817-21838
START_RANGE client112d socket1036 210 21838-21859
START_RANGE client112e socket1037 210 21859-21880
START_RANGE client112f socket1038 210 21880-21901
START_RANGE client112g socket1039 210 21901-21922
START_RANGE client112h socket1040 220 21922-21944
START_RANGE client112i socket1041 210 21944-21965
START_RANGE client112j socket1042 210 21965-21986
START_RANGE client112k socket1043 210 21986-22007
START_RANGE client112l socket1044 210 22007-22028
START_RANGE client112a socket1045 210 22028-22049
START_RANGE client112b socket1046 210 22049-22070
START_RANGE client112c socket1047 210 22070-22091
START_RANGE client112d socket1048 210 22091-22112
START_RANGE client112e socket1049 210 22112-22133
START_RANGE client112f socket1050 220 22133-22155
START_RANGE client112g socket1051 210 22155-22176
START_RANGE client112h socket1052 210 22176-22197
START_RANGE client112i socket1053 210 22197-22218
START_RANGE client112j socket1054 210 22218-22239
START_RANGE client112k socket1055 210 22239-22260
START_RANGE client112l socket1056 210 22260-22281

START_RANGE client121a socket1057 210 22281-22302
START_RANGE client121b socket1058 210 22302-22323
START_RANGE client121c socket1059 210 22323-22344
START_RANGE client121d socket1060 220 22344-22366
START_RANGE client121e socket1061 210 22366-22387
START_RANGE client121f socket1062 210 22387-22408
START_RANGE client121g socket1063 210 22408-22429
START_RANGE client121h socket1064 210 22429-22450
START_RANGE client121i socket1065 210 22450-22471
START_RANGE client121j socket1066 210 22471-22492
START_RANGE client121k socket1067 210 22492-22513
START_RANGE client121l socket1068 210 22513-22534
START_RANGE client121a socket1069 210 22534-22555
START_RANGE client121b socket1070 220 22555-22577
START_RANGE client121c socket1071 210 22577-22598
START_RANGE client121d socket1072 210 22598-22619
START_RANGE client121e socket1073 210 22619-22640
START_RANGE client121f socket1074 210 22640-22661
START_RANGE client121g socket1075 210 22661-22682
START_RANGE client121h socket1076 210 22682-22703
START_RANGE client121i socket1077 210 22703-22724
START_RANGE client121j socket1078 210 22724-22745
START_RANGE client121k socket1079 210 22745-22766
START_RANGE client121l socket1080 220 22766-22788
START_RANGE client121a socket1081 210 22788-22809
START_RANGE client121b socket1082 210 22809-22830
START_RANGE client121c socket1083 210 22830-22851
START_RANGE client121d socket1084 210 22851-22872
START_RANGE client121e socket1085 210 22872-22893
START_RANGE client121f socket1086 210 22893-22914
START_RANGE client121g socket1087 210 22914-22935
START_RANGE client121h socket1088 210 22935-22956

```
START_RANGE client121i socket1089 210 22956-22977
START_RANGE client121j socket1090 220 22977-22999
START_RANGE client121k socket1091 210 22999-23020
START_RANGE client121l socket1092 210 23020-23041
START_RANGE client121a socket1093 210 23041-23062
START_RANGE client121b socket1094 210 23062-23083
START_RANGE client121c socket1095 210 23083-23104
START_RANGE client121d socket1096 210 23104-23125
START_RANGE client121e socket1097 210 23125-23146
START_RANGE client121f socket1098 210 23146-23167
START_RANGE client121g socket1099 210 23167-23188
START_RANGE client121h socket1100 220 23188-23210
START_RANGE client121i socket1101 210 23210-23231
START_RANGE client121j socket1102 210 23231-23252
START_RANGE client121k socket1103 210 23252-23273
START_RANGE client121l socket1104 210 23273-23294

START_RANGE client122a socket1105 210 23294-23315
START_RANGE client122b socket1106 210 23315-23336
START_RANGE client122c socket1107 210 23336-23357
START_RANGE client122d socket1108 210 23357-23378
START_RANGE client122e socket1109 210 23378-23399
START_RANGE client122f socket1110 220 23399-23421
START_RANGE client122g socket1111 210 23421-23442
START_RANGE client122h socket1112 210 23442-23463
START_RANGE client122i socket1113 210 23463-23484
START_RANGE client122j socket1114 210 23484-23505
START_RANGE client122k socket1115 210 23505-23526
START_RANGE client122l socket1116 210 23526-23547
START_RANGE client122a socket1117 210 23547-23568
START_RANGE client122b socket1118 210 23568-23589
START_RANGE client122c socket1119 210 23589-23610
START_RANGE client122d socket1120 220 23610-23632
START_RANGE client122e socket1121 210 23632-23653
START_RANGE client122f socket1122 210 23653-23674
START_RANGE client122g socket1123 210 23674-23695
START_RANGE client122h socket1124 210 23695-23716
START_RANGE client122i socket1125 210 23716-23737
START_RANGE client122j socket1126 210 23737-23758
START_RANGE client122k socket1127 210 23758-23779
START_RANGE client122l socket1128 210 23779-23800
START_RANGE client122a socket1129 210 23800-23821
START_RANGE client122b socket1130 220 23821-23843
START_RANGE client122c socket1131 210 23843-23864
START_RANGE client122d socket1132 210 23864-23885
START_RANGE client122e socket1133 210 23885-23906
START_RANGE client122f socket1134 210 23906-23927
START_RANGE client122g socket1135 210 23927-23948
START_RANGE client122h socket1136 210 23948-23969
START_RANGE client122i socket1137 210 23969-23990
START_RANGE client122j socket1138 210 23990-24011
START_RANGE client122k socket1139 210 24011-24032
START_RANGE client122l socket1140 220 24032-24054
START_RANGE client122a socket1141 210 24054-24075
START_RANGE client122b socket1142 210 24075-24096
START_RANGE client122c socket1143 210 24096-24117
START_RANGE client122d socket1144 210 24117-24138
START_RANGE client122e socket1145 210 24138-24159
START_RANGE client122f socket1146 210 24159-24180
START_RANGE client122g socket1147 210 24180-24201
START_RANGE client122h socket1148 210 24201-24222
START_RANGE client122i socket1149 210 24222-24243
START_RANGE client122j socket1150 220 24243-24265
START_RANGE client122k socket1151 210 24265-24286
START_RANGE client122l socket1152 210 24286-24307

#elif MASTER_NUM5
START_RANGE client131a socket1153 210 24307-24328
START_RANGE client131b socket1154 210 24328-24349
START_RANGE client131c socket1155 210 24349-24370
START_RANGE client131d socket1156 210 24370-24391
START_RANGE client131e socket1157 210 24391-24412
START_RANGE client131f socket1158 210 24412-24433
START_RANGE client131g socket1159 210 24433-24454
START_RANGE client131h socket1160 220 24454-24476
START_RANGE client131i socket1161 210 24476-24497
START_RANGE client131j socket1162 210 24497-24518
START_RANGE client131k socket1163 210 24518-24539
START_RANGE client131l socket1164 210 24539-24560
START_RANGE client131a socket1165 210 24560-24581
START_RANGE client131b socket1166 210 24581-24602
START_RANGE client131c socket1167 210 24602-24623
START_RANGE client131d socket1168 210 24623-24644
START_RANGE client131e socket1169 210 24644-24665
START_RANGE client131f socket1170 220 24665-24687
START_RANGE client131g socket1171 210 24687-24708
START_RANGE client131h socket1172 210 24708-24729
START_RANGE client131i socket1173 210 24729-24750
START_RANGE client131j socket1174 210 24750-24771
START_RANGE client131k socket1175 210 24771-24792
START_RANGE client131l socket1176 210 24792-24813
START_RANGE client131a socket1177 210 24813-24834
START_RANGE client131b socket1178 210 24834-24855
START_RANGE client131c socket1179 210 24855-24876
START_RANGE client131d socket1180 220 24876-24898
START_RANGE client131e socket1181 210 24898-24919
START_RANGE client131f socket1182 210 24919-24940
START_RANGE client131g socket1183 210 24940-24961
START_RANGE client131h socket1184 210 24961-24982
START_RANGE client131i socket1185 210 24982-25003
START_RANGE client131j socket1186 210 25003-25024
START_RANGE client131k socket1187 210 25024-25045
START_RANGE client131l socket1188 210 25045-25066
START_RANGE client131a socket1189 210 25066-25087
START_RANGE client131b socket1190 220 25087-25109
START_RANGE client131c socket1191 210 25109-25130
START_RANGE client131d socket1192 210 25130-25151
START_RANGE client131e socket1193 210 25151-25172
START_RANGE client131f socket1194 210 25172-25193
START_RANGE client131g socket1195 210 25193-25214
START_RANGE client131h socket1196 210 25214-25235
START_RANGE client131i socket1197 210 25235-25256
START_RANGE client131j socket1198 210 25256-25277
START_RANGE client131k socket1199 210 25277-25298
START_RANGE client131l socket1200 220 25298-25320

START_RANGE client132a socket1201 210 25320-25341
START_RANGE client132b socket1202 210 25341-25362
START_RANGE client132c socket1203 210 25362-25383
START_RANGE client132d socket1204 210 25383-25404
START_RANGE client132e socket1205 210 25404-25425
START_RANGE client132f socket1206 210 25425-25446
START_RANGE client132g socket1207 210 25446-25467
START_RANGE client132h socket1208 210 25467-25488
START_RANGE client132i socket1209 210 25488-25509
START_RANGE client132j socket1210 220 25509-25531
START_RANGE client132k socket1211 210 25531-25552
START_RANGE client132l socket1212 210 25552-25573
START_RANGE client132a socket1213 210 25573-25594
START_RANGE client132b socket1214 210 25594-25615
START_RANGE client132c socket1215 210 25615-25636
START_RANGE client132d socket1216 210 25636-25657
START_RANGE client132e socket1217 210 25657-25678
START_RANGE client132f socket1218 210 25678-25699
START_RANGE client132g socket1219 210 25699-25720
START_RANGE client132h socket1220 220 25720-25742
START_RANGE client132i socket1221 210 25742-25763
START_RANGE client132j socket1222 210 25763-25784
START_RANGE client132k socket1223 210 25784-25805
START_RANGE client132l socket1224 210 25805-25826
START_RANGE client132a socket1225 210 25826-25847
START_RANGE client132b socket1226 210 25847-25868
START_RANGE client132c socket1227 210 25868-25889
START_RANGE client132d socket1228 210 25889-25910
START_RANGE client132e socket1229 210 25910-25931
START_RANGE client132f socket1230 220 25931-25953
START_RANGE client132g socket1231 210 25953-25974
START_RANGE client132h socket1232 210 25974-25995
START_RANGE client132i socket1233 210 25995-26016
START_RANGE client132j socket1234 210 26016-26037
START_RANGE client132k socket1235 210 26037-26058
START_RANGE client132l socket1236 210 26058-26079
START_RANGE client132a socket1237 210 26079-26100
START_RANGE client132b socket1238 210 26100-26121
START_RANGE client132c socket1239 210 26121-26142
START_RANGE client132d socket1240 220 26142-26164
START_RANGE client132e socket1241 210 26164-26185
START_RANGE client132f socket1242 210 26185-26206
START_RANGE client132g socket1243 210 26206-26227
START_RANGE client132h socket1244 210 26227-26248
START_RANGE client132i socket1245 210 26248-26269
START_RANGE client132j socket1246 210 26269-26290
START_RANGE client132k socket1247 210 26290-26311
START_RANGE client132l socket1248 210 26311-26332

START_RANGE client141a socket1249 210 26332-26353
START_RANGE client141b socket1250 220 26353-26375
START_RANGE client141c socket1251 210 26375-26396
START_RANGE client141d socket1252 210 26396-26417
START_RANGE client141e socket1253 210 26417-26438
START_RANGE client141f socket1254 210 26438-26459
START_RANGE client141g socket1255 210 26459-26480
START_RANGE client141h socket1256 210 26480-26501
START_RANGE client141i socket1257 210 26501-26522
START_RANGE client141j socket1258 210 26522-26543
START_RANGE client141k socket1259 210 26543-26564
START_RANGE client141l socket1260 220 26564-26586
START_RANGE client141a socket1261 210 26586-26607
START_RANGE client141b socket1262 210 26607-26628
START_RANGE client141c socket1263 210 26628-26649
START_RANGE client141d socket1264 210 26649-26670
START_RANGE client141e socket1265 210 26670-26691
START_RANGE client141f socket1266 210 26691-26712
START_RANGE client141g socket1267 210 26712-26733
START_RANGE client141h socket1268 210 26733-26754
START_RANGE client141i socket1269 210 26754-26775
START_RANGE client141j socket1270 220 26775-26797
START_RANGE client141k socket1271 210 26797-26818
START_RANGE client141l socket1272 210 26818-26839
START_RANGE client141a socket1273 210 26839-26860
START_RANGE client141b socket1274 210 26860-26881
START_RANGE client141c socket1275 210 26881-26902
START_RANGE client141d socket1276 210 26902-26923
START_RANGE client141e socket1277 210 26923-26944
START_RANGE client141f socket1278 210 26944-26965
START_RANGE client141g socket1279 210 26965-26986
START_RANGE client141h socket1280 220 26986-27008
START_RANGE client141i socket1281 210 27008-27029
```

```
START_RANGE client141j socket1282 210 27029-27050
START_RANGE client141k socket1283 210 27050-27071
START_RANGE client141l socket1284 210 27071-27092
START_RANGE client141a socket1285 210 27092-27113
START_RANGE client141b socket1286 210 27113-27134
START_RANGE client141c socket1287 210 27134-27155
START_RANGE client141d socket1288 210 27155-27176
START_RANGE client141e socket1289 210 27176-27197
START_RANGE client141f socket1290 220 27197-27219
START_RANGE client141g socket1291 210 27219-27240
START_RANGE client141h socket1292 210 27240-27261
START_RANGE client141i socket1293 210 27261-27282
START_RANGE client141j socket1294 210 27282-27303
START_RANGE client141k socket1295 210 27303-27324
START_RANGE client141l socket1296 210 27324-27345

START_RANGE client142a socket1297 210 27345-27366
START_RANGE client142b socket1298 210 27366-27387
START_RANGE client142c socket1299 210 27387-27408
START_RANGE client142d socket1300 220 27408-27430
START_RANGE client142e socket1301 210 27430-27451
START_RANGE client142f socket1302 210 27451-27472
START_RANGE client142g socket1303 210 27472-27493
START_RANGE client142h socket1304 210 27493-27514
START_RANGE client142i socket1305 210 27514-27535
START_RANGE client142j socket1306 210 27535-27556
START_RANGE client142k socket1307 210 27556-27577
START_RANGE client142l socket1308 210 27577-27598
START_RANGE client142a socket1309 210 27598-27619
START_RANGE client142b socket1310 220 27619-27641
START_RANGE client142c socket1311 210 27641-27662
START_RANGE client142d socket1312 210 27662-27683
START_RANGE client142e socket1313 210 27683-27704
START_RANGE client142f socket1314 210 27704-27725
START_RANGE client142g socket1315 210 27725-27746
START_RANGE client142h socket1316 210 27746-27767
START_RANGE client142i socket1317 210 27767-27788
START_RANGE client142j socket1318 210 27788-27809
START_RANGE client142k socket1319 210 27809-27830
START_RANGE client142l socket1320 220 27830-27852
START_RANGE client142a socket1321 210 27852-27873
START_RANGE client142b socket1322 210 27873-27894
START_RANGE client142c socket1323 210 27894-27915
START_RANGE client142d socket1324 210 27915-27936
START_RANGE client142e socket1325 210 27936-27957
START_RANGE client142f socket1326 210 27957-27978
START_RANGE client142g socket1327 210 27978-27999
START_RANGE client142h socket1328 210 27999-28020
START_RANGE client142i socket1329 210 28020-28041
START_RANGE client142j socket1330 220 28041-28063
START_RANGE client142k socket1331 210 28063-28084
START_RANGE client142l socket1332 210 28084-28105
START_RANGE client142a socket1333 210 28105-28126
START_RANGE client142b socket1334 210 28126-28147
START_RANGE client142c socket1335 210 28147-28168
START_RANGE client142d socket1336 210 28168-28189
START_RANGE client142e socket1337 210 28189-28210
START_RANGE client142f socket1338 210 28210-28231
START_RANGE client142g socket1339 210 28231-28252
START_RANGE client142h socket1340 220 28252-28274
START_RANGE client142i socket1341 210 28274-28295
START_RANGE client142j socket1342 210 28295-28316
START_RANGE client142k socket1343 210 28316-28337
START_RANGE client142l socket1344 210 28337-28358

START_RANGE client151a socket1345 210 28358-28379
START_RANGE client151b socket1346 210 28379-28400
START_RANGE client151c socket1347 210 28400-28421
START_RANGE client151d socket1348 210 28421-28442
START_RANGE client151e socket1349 210 28442-28463
START_RANGE client151f socket1350 220 28463-28485
START_RANGE client151g socket1351 210 28485-28506
START_RANGE client151h socket1352 210 28506-28527
START_RANGE client151i socket1353 210 28527-28548
START_RANGE client151j socket1354 210 28548-28569
START_RANGE client151k socket1355 210 28569-28590
START_RANGE client151l socket1356 210 28590-28611
START_RANGE client151a socket1357 210 28611-28632
START_RANGE client151b socket1358 210 28632-28653
START_RANGE client151c socket1359 210 28653-28674
START_RANGE client151d socket1360 220 28674-28696
START_RANGE client151e socket1361 210 28696-28717
START_RANGE client151f socket1362 210 28717-28738
START_RANGE client151g socket1363 210 28738-28759
START_RANGE client151h socket1364 210 28759-28780
START_RANGE client151i socket1365 210 28780-28801
START_RANGE client151j socket1366 210 28801-28822
START_RANGE client151k socket1367 210 28822-28843
START_RANGE client151l socket1368 210 28843-28864
START_RANGE client151a socket1369 210 28864-28885
START_RANGE client151b socket1370 220 28885-28907
START_RANGE client151c socket1371 210 28907-28928
START_RANGE client151d socket1372 210 28928-28949
START_RANGE client151e socket1373 210 28949-28970
START_RANGE client151f socket1374 210 28970-28991
START_RANGE client151g socket1375 210 28991-29012
START_RANGE client151h socket1376 210 29012-29033
START_RANGE client151i socket1377 210 29033-29054
START_RANGE client151j socket1378 210 29054-29075
START_RANGE client151k socket1379 210 29075-29096
START_RANGE client151l socket1380 220 29096-29118
START_RANGE client151a socket1381 210 29118-29139
START_RANGE client151b socket1382 210 29139-29160
START_RANGE client151c socket1383 210 29160-29181
START_RANGE client151d socket1384 210 29181-29202
START_RANGE client151e socket1385 210 29202-29223
START_RANGE client151f socket1386 210 29223-29244
START_RANGE client151g socket1387 210 29244-29265
START_RANGE client151h socket1388 210 29265-29286
START_RANGE client151i socket1389 210 29286-29307
START_RANGE client151j socket1390 220 29307-29329
START_RANGE client151k socket1391 210 29329-29350
START_RANGE client151l socket1392 210 29350-29371

START_RANGE client152a socket1393 210 29371-29392
START_RANGE client152b socket1394 210 29392-29413
START_RANGE client152c socket1395 210 29413-29434
START_RANGE client152d socket1396 210 29434-29455
START_RANGE client152e socket1397 210 29455-29476
START_RANGE client152f socket1398 210 29476-29497
START_RANGE client152g socket1399 210 29497-29518
START_RANGE client152h socket1400 220 29518-29540
START_RANGE client152i socket1401 210 29540-29561
START_RANGE client152j socket1402 210 29561-29582
START_RANGE client152k socket1403 210 29582-29603
START_RANGE client152l socket1404 210 29603-29624
START_RANGE client152a socket1405 210 29624-29645
START_RANGE client152b socket1406 210 29645-29666
START_RANGE client152c socket1407 210 29666-29687
START_RANGE client152d socket1408 210 29687-29708
START_RANGE client152e socket1409 210 29708-29729
START_RANGE client152f socket1410 220 29729-29751
START_RANGE client152g socket1411 210 29751-29772
START_RANGE client152h socket1412 210 29772-29793
START_RANGE client152i socket1413 210 29793-29814
START_RANGE client152j socket1414 210 29814-29835
START_RANGE client152k socket1415 210 29835-29856
START_RANGE client152l socket1416 210 29856-29877
START_RANGE client152a socket1417 210 29877-29898
START_RANGE client152b socket1418 210 29898-29919
START_RANGE client152c socket1419 210 29919-29940
START_RANGE client152d socket1420 220 29940-29962
START_RANGE client152e socket1421 210 29962-29983
START_RANGE client152f socket1422 210 29983-30004
START_RANGE client152g socket1423 210 30004-30025
START_RANGE client152h socket1424 210 30025-30046
START_RANGE client152i socket1425 210 30046-30067
START_RANGE client152j socket1426 210 30067-30088
START_RANGE client152k socket1427 210 30088-30109
START_RANGE client152l socket1428 210 30109-30130
START_RANGE client152a socket1429 210 30130-30151
START_RANGE client152b socket1430 220 30151-30173
START_RANGE client152c socket1431 210 30173-30194
START_RANGE client152d socket1432 210 30194-30215
START_RANGE client152e socket1433 210 30215-30236
START_RANGE client152f socket1434 210 30236-30257
START_RANGE client152g socket1435 210 30257-30278
START_RANGE client152h socket1436 210 30278-30299
START_RANGE client152i socket1437 210 30299-30320
START_RANGE client152j socket1438 210 30320-30341
START_RANGE client152k socket1439 210 30341-30362
START_RANGE client152l socket1440 220 30362-30384

#elif MASTER_NUM6
START_RANGE client161a socket1441 210 30384-30405
START_RANGE client161b socket1442 210 30405-30426
START_RANGE client161c socket1443 210 30426-30447
START_RANGE client161d socket1444 210 30447-30468
START_RANGE client161e socket1445 210 30468-30489
START_RANGE client161f socket1446 210 30489-30510
START_RANGE client161g socket1447 210 30510-30531
START_RANGE client161h socket1448 210 30531-30552
START_RANGE client161i socket1449 210 30552-30573
START_RANGE client161j socket1450 220 30573-30595
START_RANGE client161k socket1451 210 30595-30616
START_RANGE client161l socket1452 210 30616-30637
START_RANGE client161a socket1453 210 30637-30658
START_RANGE client161b socket1454 210 30658-30679
START_RANGE client161c socket1455 210 30679-30700
START_RANGE client161d socket1456 210 30700-30721
START_RANGE client161e socket1457 210 30721-30742
START_RANGE client161f socket1458 210 30742-30763
START_RANGE client161g socket1459 210 30763-30784
START_RANGE client161h socket1460 220 30784-30806
START_RANGE client161i socket1461 210 30806-30827
START_RANGE client161j socket1462 210 30827-30848
START_RANGE client161k socket1463 210 30848-30869
START_RANGE client161l socket1464 210 30869-30890
START_RANGE client161a socket1465 210 30890-30911
START_RANGE client161b socket1466 210 30911-30932
START_RANGE client161c socket1467 210 30932-30953
START_RANGE client161d socket1468 210 30953-30974
START_RANGE client161e socket1469 210 30974-30995
START_RANGE client161f socket1470 220 30995-31017
START_RANGE client161g socket1471 210 31017-31038
START_RANGE client161h socket1472 210 31038-31059
START_RANGE client161i socket1473 210 31059-31080
START_RANGE client161j socket1474 210 31080-31101
```

START_RANGE client161k socket1475 210 31101-31122
START_RANGE client161l socket1476 210 31122-31143
START_RANGE client161a socket1477 210 31143-31164
START_RANGE client161b socket1478 210 31164-31185
START_RANGE client161c socket1479 210 31185-31206
START_RANGE client161d socket1480 220 31206-31228
START_RANGE client161e socket1481 210 31228-31249
START_RANGE client161f socket1482 210 31249-31270
START_RANGE client161g socket1483 210 31270-31291
START_RANGE client161h socket1484 210 31291-31312
START_RANGE client161i socket1485 210 31312-31333
START_RANGE client161j socket1486 210 31333-31354
START_RANGE client161k socket1487 210 31354-31375
START_RANGE client161l socket1488 210 31375-31396

START_RANGE client162a socket1489 210 31396-31417
START_RANGE client162b socket1490 220 31417-31439
START_RANGE client162c socket1491 210 31439-31460
START_RANGE client162d socket1492 210 31460-31481
START_RANGE client162e socket1493 210 31481-31502
START_RANGE client162f socket1494 210 31502-31523
START_RANGE client162g socket1495 210 31523-31544
START_RANGE client162h socket1496 210 31544-31565
START_RANGE client162i socket1497 210 31565-31586
START_RANGE client162j socket1498 210 31586-31607
START_RANGE client162k socket1499 210 31607-31628
START_RANGE client162l socket1500 220 31628-31650
START_RANGE client162a socket1501 210 31650-31671
START_RANGE client162b socket1502 210 31671-31692
START_RANGE client162c socket1503 210 31692-31713
START_RANGE client162d socket1504 210 31713-31734
START_RANGE client162e socket1505 210 31734-31755
START_RANGE client162f socket1506 210 31755-31776
START_RANGE client162g socket1507 210 31776-31797
START_RANGE client162h socket1508 210 31797-31818
START_RANGE client162i socket1509 210 31818-31839
START_RANGE client162j socket1510 220 31839-31861
START_RANGE client162k socket1511 210 31861-31882
START_RANGE client162l socket1512 210 31882-31903
START_RANGE client162a socket1513 210 31903-31924
START_RANGE client162b socket1514 210 31924-31945
START_RANGE client162c socket1515 210 31945-31966
START_RANGE client162d socket1516 210 31966-31987
START_RANGE client162e socket1517 210 31987-32008
START_RANGE client162f socket1518 210 32008-32029
START_RANGE client162g socket1519 210 32029-32050
START_RANGE client162h socket1520 220 32050-32072
START_RANGE client162i socket1521 210 32072-32093
START_RANGE client162j socket1522 210 32093-32114
START_RANGE client162k socket1523 210 32114-32135
START_RANGE client162l socket1524 210 32135-32156
START_RANGE client162a socket1525 210 32156-32177
START_RANGE client162b socket1526 210 32177-32198
START_RANGE client162c socket1527 210 32198-32219
START_RANGE client162d socket1528 210 32219-32240
START_RANGE client162e socket1529 210 32240-32261
START_RANGE client162f socket1530 220 32261-32283
START_RANGE client162g socket1531 210 32283-32304
START_RANGE client162h socket1532 210 32304-32325
START_RANGE client162i socket1533 210 32325-32346
START_RANGE client162j socket1534 210 32346-32367
START_RANGE client162k socket1535 210 32367-32388
START_RANGE client162l socket1536 210 32388-32409

START_RANGE client171a socket1537 210 32409-32430
START_RANGE client171b socket1538 210 32430-32451
START_RANGE client171c socket1539 210 32451-32472
START_RANGE client171d socket1540 220 32472-32494
START_RANGE client171e socket1541 210 32494-32515
START_RANGE client171f socket1542 210 32515-32536
START_RANGE client171g socket1543 210 32536-32557
START_RANGE client171h socket1544 210 32557-32578
START_RANGE client171i socket1545 210 32578-32599
START_RANGE client171j socket1546 210 32599-32620
START_RANGE client171k socket1547 210 32620-32641
START_RANGE client171l socket1548 210 32641-32662
START_RANGE client171a socket1549 210 32662-32683
START_RANGE client171b socket1550 220 32683-32705
START_RANGE client171c socket1551 210 32705-32726
START_RANGE client171d socket1552 210 32726-32747
START_RANGE client171e socket1553 210 32747-32768
START_RANGE client171f socket1554 210 32768-32789
START_RANGE client171g socket1555 210 32789-32810
START_RANGE client171h socket1556 210 32810-32831
START_RANGE client171i socket1557 210 32831-32852
START_RANGE client171j socket1558 210 32852-32873
START_RANGE client171k socket1559 210 32873-32894
START_RANGE client171l socket1560 220 32894-32916
START_RANGE client171a socket1561 210 32916-32937
START_RANGE client171b socket1562 210 32937-32958
START_RANGE client171c socket1563 210 32958-32979
START_RANGE client171d socket1564 210 32979-33000
START_RANGE client171e socket1565 210 33000-33021
START_RANGE client171f socket1566 210 33021-33042
START_RANGE client171g socket1567 210 33042-33063
START_RANGE client171h socket1568 210 33063-33084
START_RANGE client171i socket1569 210 33084-33105
START_RANGE client171j socket1570 220 33105-33127
START_RANGE client171k socket1571 210 33127-33148
START_RANGE client171l socket1572 210 33148-33169
START_RANGE client171a socket1573 210 33169-33190
START_RANGE client171b socket1574 210 33190-33211
START_RANGE client171c socket1575 210 33211-33232
START_RANGE client171d socket1576 210 33232-33253
START_RANGE client171e socket1577 210 33253-33274
START_RANGE client171f socket1578 210 33274-33295
START_RANGE client171g socket1579 210 33295-33316
START_RANGE client171h socket1580 220 33316-33338
START_RANGE client171i socket1581 210 33338-33359
START_RANGE client171j socket1582 210 33359-33380
START_RANGE client171k socket1583 210 33380-33401
START_RANGE client171l socket1584 210 33401-33422

START_RANGE client172a socket1585 210 33422-33443
START_RANGE client172b socket1586 210 33443-33464
START_RANGE client172c socket1587 210 33464-33485
START_RANGE client172d socket1588 210 33485-33506
START_RANGE client172e socket1589 210 33506-33527
START_RANGE client172f socket1590 220 33527-33549
START_RANGE client172g socket1591 210 33549-33570
START_RANGE client172h socket1592 210 33570-33591
START_RANGE client172i socket1593 210 33591-33612
START_RANGE client172j socket1594 210 33612-33633
START_RANGE client172k socket1595 210 33633-33654
START_RANGE client172l socket1596 210 33654-33675
START_RANGE client172a socket1597 210 33675-33696
START_RANGE client172b socket1598 210 33696-33717
START_RANGE client172c socket1599 210 33717-33738
START_RANGE client172d socket1600 220 33738-33760
START_RANGE client172e socket1601 210 33760-33781
START_RANGE client172f socket1602 210 33781-33802
START_RANGE client172g socket1603 210 33802-33823
START_RANGE client172h socket1604 210 33823-33844
START_RANGE client172i socket1605 210 33844-33865
START_RANGE client172j socket1606 210 33865-33886
START_RANGE client172k socket1607 210 33886-33907
START_RANGE client172l socket1608 210 33907-33928
START_RANGE client172a socket1609 210 33928-33949
START_RANGE client172b socket1610 220 33949-33971
START_RANGE client172c socket1611 210 33971-33992
START_RANGE client172d socket1612 210 33992-34013
START_RANGE client172e socket1613 210 34013-34034
START_RANGE client172f socket1614 210 34034-34055
START_RANGE client172g socket1615 210 34055-34076
START_RANGE client172h socket1616 210 34076-34097
START_RANGE client172i socket1617 210 34097-34118
START_RANGE client172j socket1618 210 34118-34139
START_RANGE client172k socket1619 210 34139-34160
START_RANGE client172l socket1620 220 34160-34182
START_RANGE client172a socket1621 210 34182-34203
START_RANGE client172b socket1622 210 34203-34224
START_RANGE client172c socket1623 210 34224-34245
START_RANGE client172d socket1624 210 34245-34266
START_RANGE client172e socket1625 210 34266-34287
START_RANGE client172f socket1626 210 34287-34308
START_RANGE client172g socket1627 210 34308-34329
START_RANGE client172h socket1628 210 34329-34350
START_RANGE client172i socket1629 210 34350-34371
START_RANGE client172j socket1630 220 34371-34393
START_RANGE client172k socket1631 210 34393-34414
START_RANGE client172l socket1632 210 34414-34435

START_RANGE client181a socket1633 210 34435-34456
START_RANGE client181b socket1634 210 34456-34477
START_RANGE client181c socket1635 210 34477-34498
START_RANGE client181d socket1636 210 34498-34519
START_RANGE client181e socket1637 210 34519-34540
START_RANGE client181f socket1638 210 34540-34561
START_RANGE client181g socket1639 210 34561-34582
START_RANGE client181h socket1640 220 34582-34604
START_RANGE client181i socket1641 210 34604-34625
START_RANGE client181j socket1642 210 34625-34646
START_RANGE client181k socket1643 210 34646-34667
START_RANGE client181l socket1644 210 34667-34688
START_RANGE client181a socket1645 210 34688-34709
START_RANGE client181b socket1646 210 34709-34730
START_RANGE client181c socket1647 210 34730-34751
START_RANGE client181d socket1648 210 34751-34772
START_RANGE client181e socket1649 210 34772-34793
START_RANGE client181f socket1650 220 34793-34815
START_RANGE client181g socket1651 210 34815-34836
START_RANGE client181h socket1652 210 34836-34857
START_RANGE client181i socket1653 210 34857-34878
START_RANGE client181j socket1654 210 34878-34899
START_RANGE client181k socket1655 210 34899-34920
START_RANGE client181l socket1656 210 34920-34941
START_RANGE client181a socket1657 210 34941-34962
START_RANGE client181b socket1658 210 34962-34983
START_RANGE client181c socket1659 210 34983-35004
START_RANGE client181d socket1660 220 35004-35026
START_RANGE client181e socket1661 210 35026-35047
START_RANGE client181f socket1662 210 35047-35068
START_RANGE client181g socket1663 210 35068-35089
START_RANGE client181h socket1664 210 35089-35110
START_RANGE client181i socket1665 210 35110-35131
START_RANGE client181j socket1666 210 35131-35152
START_RANGE client181k socket1667 210 35152-35173
START_RANGE client181l socket1668 210 35173-35194

START_RANGE client181a socket1669 210 35194-35215
START_RANGE client181b socket1670 220 35215-35237
START_RANGE client181c socket1671 210 35237-35258
START_RANGE client181d socket1672 210 35258-35279
START_RANGE client181e socket1673 210 35279-35300
START_RANGE client181f socket1674 210 35300-35321
START_RANGE client181g socket1675 210 35321-35342
START_RANGE client181h socket1676 210 35342-35363
START_RANGE client181i socket1677 210 35363-35384
START_RANGE client181j socket1678 210 35384-35405
START_RANGE client181k socket1679 210 35405-35426
START_RANGE client181l socket1680 220 35426-35448

START_RANGE client182a socket1681 210 35448-35469
START_RANGE client182b socket1682 210 35469-35490
START_RANGE client182c socket1683 210 35490-35511
START_RANGE client182d socket1684 210 35511-35532
START_RANGE client182e socket1685 210 35532-35553
START_RANGE client182f socket1686 210 35553-35574
START_RANGE client182g socket1687 210 35574-35595
START_RANGE client182h socket1688 210 35595-35616
START_RANGE client182i socket1689 210 35616-35637
START_RANGE client182j socket1690 220 35637-35659
START_RANGE client182k socket1691 210 35659-35680
START_RANGE client182l socket1692 210 35680-35701
START_RANGE client182a socket1693 210 35701-35722
START_RANGE client182b socket1694 210 35722-35743
START_RANGE client182c socket1695 210 35743-35764
START_RANGE client182d socket1696 210 35764-35785
START_RANGE client182e socket1697 210 35785-35806
START_RANGE client182f socket1698 210 35806-35827
START_RANGE client182g socket1699 210 35827-35848
START_RANGE client182h socket1700 220 35848-35870
START_RANGE client182i socket1701 210 35870-35891
START_RANGE client182j socket1702 210 35891-35912
START_RANGE client182k socket1703 210 35912-35933
START_RANGE client182l socket1704 210 35933-35954
START_RANGE client182a socket1705 210 35954-35975
START_RANGE client182b socket1706 210 35975-35996
START_RANGE client182c socket1707 210 35996-36017
START_RANGE client182d socket1708 210 36017-36038
START_RANGE client182e socket1709 210 36038-36059
START_RANGE client182f socket1710 220 36059-36081
START_RANGE client182g socket1711 210 36081-36102
START_RANGE client182h socket1712 210 36102-36123
START_RANGE client182i socket1713 210 36123-36144
START_RANGE client182j socket1714 210 36144-36165
START_RANGE client182k socket1715 210 36165-36186
START_RANGE client182l socket1716 210 36186-36207
START_RANGE client182a socket1717 210 36207-36228
START_RANGE client182b socket1718 210 36228-36249
START_RANGE client182c socket1719 210 36249-36270
START_RANGE client182d socket1720 220 36270-36292
START_RANGE client182e socket1721 210 36292-36313
START_RANGE client182f socket1722 210 36313-36334
START_RANGE client182g socket1723 210 36334-36355
START_RANGE client182h socket1724 210 36355-36376
START_RANGE client182i socket1725 210 36376-36397
START_RANGE client182j socket1726 210 36397-36418
START_RANGE client182k socket1727 210 36418-36439
START_RANGE client182l socket1728 210 36439-36460

#elif MASTER_NUM7
START_RANGE client191a socket1729 210 36460-36481
START_RANGE client191b socket1730 220 36481-36503
START_RANGE client191c socket1731 210 36503-36524
START_RANGE client191d socket1732 210 36524-36545
START_RANGE client191e socket1733 210 36545-36566
START_RANGE client191f socket1734 210 36566-36587
START_RANGE client191g socket1735 210 36587-36608
START_RANGE client191h socket1736 210 36608-36629
START_RANGE client191i socket1737 210 36629-36650
START_RANGE client191j socket1738 210 36650-36671
START_RANGE client191k socket1739 210 36671-36692
START_RANGE client191l socket1740 220 36692-36714
START_RANGE client191a socket1741 210 36714-36735
START_RANGE client191b socket1742 210 36735-36756
START_RANGE client191c socket1743 210 36756-36777
START_RANGE client191d socket1744 210 36777-36798
START_RANGE client191e socket1745 210 36798-36819
START_RANGE client191f socket1746 210 36819-36840
START_RANGE client191g socket1747 210 36840-36861
START_RANGE client191h socket1748 210 36861-36882
START_RANGE client191i socket1749 210 36882-36903
START_RANGE client191j socket1750 220 36903-36925
START_RANGE client191k socket1751 210 36925-36946
START_RANGE client191l socket1752 210 36946-36967
START_RANGE client191a socket1753 210 36967-36988
START_RANGE client191b socket1754 210 36988-37009
START_RANGE client191c socket1755 210 37009-37030
START_RANGE client191d socket1756 210 37030-37051
START_RANGE client191e socket1757 210 37051-37072
START_RANGE client191f socket1758 210 37072-37093
START_RANGE client191g socket1759 210 37093-37114
START_RANGE client191h socket1760 220 37114-37136
START_RANGE client191i socket1761 210 37136-37157
START_RANGE client191j socket1762 210 37157-37178
START_RANGE client191k socket1763 210 37178-37199
START_RANGE client191l socket1764 210 37199-37220

START_RANGE client191a socket1765 210 37220-37241
START_RANGE client191b socket1766 210 37241-37262
START_RANGE client191c socket1767 210 37262-37283
START_RANGE client191d socket1768 210 37283-37304
START_RANGE client191e socket1769 210 37304-37325
START_RANGE client191f socket1770 220 37325-37347
START_RANGE client191g socket1771 210 37347-37368
START_RANGE client191h socket1772 210 37368-37389
START_RANGE client191i socket1773 210 37389-37410
START_RANGE client191j socket1774 210 37410-37431
START_RANGE client191k socket1775 210 37431-37452
START_RANGE client191l socket1776 210 37452-37473

START_RANGE client192a socket1777 210 37473-37494
START_RANGE client192b socket1778 210 37494-37515
START_RANGE client192c socket1779 210 37515-37536
START_RANGE client192d socket1780 220 37536-37558
START_RANGE client192e socket1781 210 37558-37579
START_RANGE client192f socket1782 210 37579-37600
START_RANGE client192g socket1783 210 37600-37621
START_RANGE client192h socket1784 210 37621-37642
START_RANGE client192i socket1785 210 37642-37663
START_RANGE client192j socket1786 210 37663-37684
START_RANGE client192k socket1787 210 37684-37705
START_RANGE client192l socket1788 210 37705-37726
START_RANGE client192a socket1789 210 37726-37747
START_RANGE client192b socket1790 220 37747-37769
START_RANGE client192c socket1791 210 37769-37790
START_RANGE client192d socket1792 210 37790-37811
START_RANGE client192e socket1793 210 37811-37832
START_RANGE client192f socket1794 210 37832-37853
START_RANGE client192g socket1795 210 37853-37874
START_RANGE client192h socket1796 210 37874-37895
START_RANGE client192i socket1797 210 37895-37916
START_RANGE client192j socket1798 210 37916-37937
START_RANGE client192k socket1799 210 37937-37958
START_RANGE client192l socket1800 220 37958-37980
START_RANGE client192a socket1801 210 37980-38001
START_RANGE client192b socket1802 210 38001-38022
START_RANGE client192c socket1803 210 38022-38043
START_RANGE client192d socket1804 210 38043-38064
START_RANGE client192e socket1805 210 38064-38085
START_RANGE client192f socket1806 210 38085-38106
START_RANGE client192g socket1807 210 38106-38127
START_RANGE client192h socket1808 210 38127-38148
START_RANGE client192i socket1809 210 38148-38169
START_RANGE client192j socket1810 220 38169-38191
START_RANGE client192k socket1811 210 38191-38212
START_RANGE client192l socket1812 210 38212-38233
START_RANGE client192a socket1813 210 38233-38254
START_RANGE client192b socket1814 210 38254-38275
START_RANGE client192c socket1815 210 38275-38296
START_RANGE client192d socket1816 210 38296-38317
START_RANGE client192e socket1817 210 38317-38338
START_RANGE client192f socket1818 210 38338-38359
START_RANGE client192g socket1819 210 38359-38380
START_RANGE client192h socket1820 220 38380-38402
START_RANGE client192i socket1821 210 38402-38423
START_RANGE client192j socket1822 210 38423-38444
START_RANGE client192k socket1823 210 38444-38465
START_RANGE client192l socket1824 210 38465-38486

START_RANGE client201a socket1825 210 38486-38507
START_RANGE client201b socket1826 210 38507-38528
START_RANGE client201c socket1827 210 38528-38549
START_RANGE client201d socket1828 210 38549-38570
START_RANGE client201e socket1829 210 38570-38591
START_RANGE client201f socket1830 220 38591-38613
START_RANGE client201g socket1831 210 38613-38634
START_RANGE client201h socket1832 210 38634-38655
START_RANGE client201i socket1833 210 38655-38676
START_RANGE client201j socket1834 210 38676-38697
START_RANGE client201k socket1835 210 38697-38718
START_RANGE client201l socket1836 210 38718-38739
START_RANGE client201a socket1837 210 38739-38760
START_RANGE client201b socket1838 210 38760-38781
START_RANGE client201c socket1839 210 38781-38802
START_RANGE client201d socket1840 220 38802-38824
START_RANGE client201e socket1841 210 38824-38845
START_RANGE client201f socket1842 210 38845-38866
START_RANGE client201g socket1843 210 38866-38887
START_RANGE client201h socket1844 210 38887-38908
START_RANGE client201i socket1845 210 38908-38929
START_RANGE client201j socket1846 210 38929-38950
START_RANGE client201k socket1847 210 38950-38971
START_RANGE client201l socket1848 210 38971-38992
START_RANGE client201a socket1849 210 38992-39013
START_RANGE client201b socket1850 220 39013-39035
START_RANGE client201c socket1851 210 39035-39056
START_RANGE client201d socket1852 210 39056-39077
START_RANGE client201e socket1853 210 39077-39098
START_RANGE client201f socket1854 210 39098-39119
START_RANGE client201g socket1855 210 39119-39140
START_RANGE client201h socket1856 210 39140-39161
START_RANGE client201i socket1857 210 39161-39182
START_RANGE client201j socket1858 210 39182-39203
START_RANGE client201k socket1859 210 39203-39224
START_RANGE client201l socket1860 220 39224-39246
START_RANGE client201a socket1861 210 39246-39267

```
START_RANGE client201b socket1862 210 39267-39288
START_RANGE client201c socket1863 210 39288-39309
START_RANGE client201d socket1864 210 39309-39330
START_RANGE client201e socket1865 210 39330-39351
START_RANGE client201f socket1866 210 39351-39372
START_RANGE client201g socket1867 210 39372-39393
START_RANGE client201h socket1868 210 39393-39414
START_RANGE client201i socket1869 210 39414-39435
START_RANGE client201j socket1870 220 39435-39457
START_RANGE client201k socket1871 210 39457-39478
START_RANGE client201l socket1872 210 39478-39499

START_RANGE client202a socket1873 210 39499-39520
START_RANGE client202b socket1874 210 39520-39541
START_RANGE client202c socket1875 210 39541-39562
START_RANGE client202d socket1876 210 39562-39583
START_RANGE client202e socket1877 210 39583-39604
START_RANGE client202f socket1878 210 39604-39625
START_RANGE client202g socket1879 210 39625-39646
START_RANGE client202h socket1880 220 39646-39668
START_RANGE client202i socket1881 210 39668-39689
START_RANGE client202j socket1882 210 39689-39710
START_RANGE client202k socket1883 210 39710-39731
START_RANGE client202l socket1884 210 39731-39752
START_RANGE client202a socket1885 210 39752-39773
START_RANGE client202b socket1886 210 39773-39794
START_RANGE client202c socket1887 210 39794-39815
START_RANGE client202d socket1888 210 39815-39836
START_RANGE client202e socket1889 210 39836-39857
START_RANGE client202f socket1890 220 39857-39879
START_RANGE client202g socket1891 210 39879-39900
START_RANGE client202h socket1892 210 39900-39921
START_RANGE client202i socket1893 210 39921-39942
START_RANGE client202j socket1894 210 39942-39963
START_RANGE client202k socket1895 210 39963-39984
START_RANGE client202l socket1896 210 39984-40005
START_RANGE client202a socket1897 210 40005-40026
START_RANGE client202b socket1898 210 40026-40047
START_RANGE client202c socket1899 210 40047-40068
START_RANGE client202d socket1900 220 40068-40090
START_RANGE client202e socket1901 210 40090-40111
START_RANGE client202f socket1902 210 40111-40132
START_RANGE client202g socket1903 210 40132-40153
START_RANGE client202h socket1904 210 40153-40174
START_RANGE client202i socket1905 210 40174-40195
START_RANGE client202j socket1906 210 40195-40216
START_RANGE client202k socket1907 210 40216-40237
START_RANGE client202l socket1908 210 40237-40258
START_RANGE client202a socket1909 210 40258-40279
START_RANGE client202b socket1910 220 40279-40301
START_RANGE client202c socket1911 210 40301-40322
START_RANGE client202d socket1912 210 40322-40343
START_RANGE client202e socket1913 210 40343-40364
START_RANGE client202f socket1914 210 40364-40385
START_RANGE client202g socket1915 210 40385-40406
START_RANGE client202h socket1916 210 40406-40427
START_RANGE client202i socket1917 210 40427-40448
START_RANGE client202j socket1918 210 40448-40469
START_RANGE client202k socket1919 210 40469-40490
START_RANGE client202l socket1920 220 40490-40512

START_RANGE client211a socket1921 210 40512-40533
START_RANGE client211b socket1922 210 40533-40554
START_RANGE client211c socket1923 210 40554-40575
START_RANGE client211d socket1924 210 40575-40596
START_RANGE client211e socket1925 210 40596-40617
START_RANGE client211f socket1926 210 40617-40638
START_RANGE client211g socket1927 210 40638-40659
START_RANGE client211h socket1928 210 40659-40680
START_RANGE client211i socket1929 210 40680-40701
START_RANGE client211j socket1930 220 40701-40723
START_RANGE client211k socket1931 210 40723-40744
START_RANGE client211l socket1932 210 40744-40765
START_RANGE client211a socket1933 210 40765-40786
START_RANGE client211b socket1934 210 40786-40807
START_RANGE client211c socket1935 210 40807-40828
START_RANGE client211d socket1936 210 40828-40849
START_RANGE client211e socket1937 210 40849-40870
START_RANGE client211f socket1938 210 40870-40891
START_RANGE client211g socket1939 210 40891-40912
START_RANGE client211h socket1940 220 40912-40934
START_RANGE client211i socket1941 210 40934-40955
START_RANGE client211j socket1942 210 40955-40976
START_RANGE client211k socket1943 210 40976-40997
START_RANGE client211l socket1944 210 40997-41018
START_RANGE client211a socket1945 210 41018-41039
START_RANGE client211b socket1946 210 41039-41060
START_RANGE client211c socket1947 210 41060-41081
START_RANGE client211d socket1948 210 41081-41102
START_RANGE client211e socket1949 210 41102-41123
START_RANGE client211f socket1950 220 41123-41145
START_RANGE client211g socket1951 210 41145-41166
START_RANGE client211h socket1952 210 41166-41187
START_RANGE client211i socket1953 210 41187-41208
START_RANGE client211j socket1954 210 41208-41229
START_RANGE client211k socket1955 210 41229-41250
START_RANGE client211l socket1956 210 41250-41271
START_RANGE client211a socket1957 210 41271-41292
START_RANGE client211b socket1958 210 41292-41313
START_RANGE client211c socket1959 210 41313-41334
START_RANGE client211d socket1960 220 41334-41356
START_RANGE client211e socket1961 210 41356-41377
START_RANGE client211f socket1962 210 41377-41398
START_RANGE client211g socket1963 210 41398-41419
START_RANGE client211h socket1964 210 41419-41440
START_RANGE client211i socket1965 210 41440-41461
START_RANGE client211j socket1966 210 41461-41482
START_RANGE client211k socket1967 210 41482-41503
START_RANGE client211l socket1968 210 41503-41524

START_RANGE client212a socket1969 210 41524-41545
START_RANGE client212b socket1970 220 41545-41567
START_RANGE client212c socket1971 210 41567-41588
START_RANGE client212d socket1972 210 41588-41609
START_RANGE client212e socket1973 210 41609-41630
START_RANGE client212f socket1974 210 41630-41651
START_RANGE client212g socket1975 210 41651-41672
START_RANGE client212h socket1976 210 41672-41693
START_RANGE client212i socket1977 210 41693-41714
START_RANGE client212j socket1978 210 41714-41735
START_RANGE client212k socket1979 210 41735-41756
START_RANGE client212l socket1980 220 41756-41778
START_RANGE client212a socket1981 210 41778-41799
START_RANGE client212b socket1982 210 41799-41820
START_RANGE client212c socket1983 210 41820-41841
START_RANGE client212d socket1984 210 41841-41862
START_RANGE client212e socket1985 210 41862-41883
START_RANGE client212f socket1986 210 41883-41904
START_RANGE client212g socket1987 210 41904-41925
START_RANGE client212h socket1988 210 41925-41946
START_RANGE client212i socket1989 210 41946-41967
START_RANGE client212j socket1990 220 41967-41989
START_RANGE client212k socket1991 210 41989-42010
START_RANGE client212l socket1992 210 42010-42031
START_RANGE client212a socket1993 210 42031-42052
START_RANGE client212b socket1994 210 42052-42073
START_RANGE client212c socket1995 210 42073-42094
START_RANGE client212d socket1996 210 42094-42115
START_RANGE client212e socket1997 210 42115-42136
START_RANGE client212f socket1998 210 42136-42157
START_RANGE client212g socket1999 210 42157-42178
START_RANGE client212h socket2000 220 42178-42200
START_RANGE client212i socket2001 210 42200-42221
START_RANGE client212j socket2002 210 42221-42242
START_RANGE client212k socket2003 210 42242-42263
START_RANGE client212l socket2004 210 42263-42284
START_RANGE client212a socket2005 210 42284-42305
START_RANGE client212b socket2006 210 42305-42326
START_RANGE client212c socket2007 210 42326-42347
START_RANGE client212d socket2008 210 42347-42368
START_RANGE client212e socket2009 210 42368-42389
START_RANGE client212f socket2010 220 42389-42411
START_RANGE client212g socket2011 210 42411-42432
START_RANGE client212h socket2012 210 42432-42453
START_RANGE client212i socket2013 210 42453-42474
START_RANGE client212j socket2014 210 42474-42495
START_RANGE client212k socket2015 210 42495-42516
START_RANGE client212l socket2016 210 42516-42537

#elif MASTER_NUM8
START_RANGE client221a socket2017 210 42537-42558
START_RANGE client221b socket2018 210 42558-42579
START_RANGE client221c socket2019 210 42579-42600
START_RANGE client221d socket2020 220 42600-42622
START_RANGE client221e socket2021 210 42622-42643
START_RANGE client221f socket2022 210 42643-42664
START_RANGE client221g socket2023 210 42664-42685
START_RANGE client221h socket2024 210 42685-42706
START_RANGE client221i socket2025 210 42706-42727
START_RANGE client221j socket2026 210 42727-42748
START_RANGE client221k socket2027 210 42748-42769
START_RANGE client221l socket2028 210 42769-42790
START_RANGE client221a socket2029 210 42790-42811
START_RANGE client221b socket2030 220 42811-42833
START_RANGE client221c socket2031 210 42833-42854
START_RANGE client221d socket2032 210 42854-42875
START_RANGE client221e socket2033 210 42875-42896
START_RANGE client221f socket2034 210 42896-42917
START_RANGE client221g socket2035 210 42917-42938
START_RANGE client221h socket2036 210 42938-42959
START_RANGE client221i socket2037 210 42959-42980
START_RANGE client221j socket2038 210 42980-43001
START_RANGE client221k socket2039 210 43001-43022
START_RANGE client221l socket2040 220 43022-43044
START_RANGE client221a socket2041 210 43044-43065
START_RANGE client221b socket2042 210 43065-43086
START_RANGE client221c socket2043 210 43086-43107
START_RANGE client221d socket2044 210 43107-43128
START_RANGE client221e socket2045 210 43128-43149
START_RANGE client221f socket2046 210 43149-43170
START_RANGE client221g socket2047 210 43170-43191
START_RANGE client221h socket2048 210 43191-43212
START_RANGE client221i socket2049 210 43212-43233
START_RANGE client221j socket2050 220 43233-43255
START_RANGE client221k socket2051 210 43255-43276
START_RANGE client221l socket2052 210 43276-43297
START_RANGE client221a socket2053 210 43297-43318
START_RANGE client221b socket2054 210 43318-43339
```

START_RANGE client221c socket2055 210 43339-43360
START_RANGE client221d socket2056 210 43360-43381
START_RANGE client221e socket2057 210 43381-43402
START_RANGE client221f socket2058 210 43402-43423
START_RANGE client221g socket2059 210 43423-43444
START_RANGE client221h socket2060 220 43444-43466
START_RANGE client221i socket2061 210 43466-43487
START_RANGE client221j socket2062 210 43487-43508
START_RANGE client221k socket2063 210 43508-43529
START_RANGE client221l socket2064 210 43529-43550

START_RANGE client222a socket2065 210 43550-43571
START_RANGE client222b socket2066 210 43571-43592
START_RANGE client222c socket2067 210 43592-43613
START_RANGE client222d socket2068 210 43613-43634
START_RANGE client222e socket2069 210 43634-43655
START_RANGE client222f socket2070 220 43655-43677
START_RANGE client222g socket2071 210 43677-43698
START_RANGE client222h socket2072 210 43698-43719
START_RANGE client222i socket2073 210 43719-43740
START_RANGE client222j socket2074 210 43740-43761
START_RANGE client222k socket2075 210 43761-43782
START_RANGE client222l socket2076 210 43782-43803
START_RANGE client222a socket2077 210 43803-43824
START_RANGE client222b socket2078 210 43824-43845
START_RANGE client222c socket2079 210 43845-43866
START_RANGE client222d socket2080 220 43866-43888
START_RANGE client222e socket2081 210 43888-43909
START_RANGE client222f socket2082 210 43909-43930
START_RANGE client222g socket2083 210 43930-43951
START_RANGE client222h socket2084 210 43951-43972
START_RANGE client222i socket2085 210 43972-43993
START_RANGE client222j socket2086 210 43993-44014
START_RANGE client222k socket2087 210 44014-44035
START_RANGE client222l socket2088 210 44035-44056
START_RANGE client222a socket2089 210 44056-44077
START_RANGE client222b socket2090 220 44077-44099
START_RANGE client222c socket2091 210 44099-44120
START_RANGE client222d socket2092 210 44120-44141
START_RANGE client222e socket2093 210 44141-44162
START_RANGE client222f socket2094 210 44162-44183
START_RANGE client222g socket2095 210 44183-44204
START_RANGE client222h socket2096 210 44204-44225
START_RANGE client222i socket2097 210 44225-44246
START_RANGE client222j socket2098 210 44246-44267
START_RANGE client222k socket2099 210 44267-44288
START_RANGE client222l socket2100 220 44288-44310
START_RANGE client222a socket2101 210 44310-44331
START_RANGE client222b socket2102 210 44331-44352
START_RANGE client222c socket2103 210 44352-44373
START_RANGE client222d socket2104 210 44373-44394
START_RANGE client222e socket2105 210 44394-44415
START_RANGE client222f socket2106 210 44415-44436
START_RANGE client222g socket2107 210 44436-44457
START_RANGE client222h socket2108 210 44457-44478
START_RANGE client222i socket2109 210 44478-44499
START_RANGE client222j socket2110 220 44499-44521
START_RANGE client222k socket2111 210 44521-44542
START_RANGE client222l socket2112 210 44542-44563

START_RANGE client231a socket2113 210 44563-44584
START_RANGE client231b socket2114 210 44584-44605
START_RANGE client231c socket2115 210 44605-44626
START_RANGE client231d socket2116 210 44626-44647
START_RANGE client231e socket2117 210 44647-44668
START_RANGE client231f socket2118 210 44668-44689
START_RANGE client231g socket2119 210 44689-44710
START_RANGE client231h socket2120 220 44710-44732
START_RANGE client231i socket2121 210 44732-44753
START_RANGE client231j socket2122 210 44753-44774
START_RANGE client231k socket2123 210 44774-44795
START_RANGE client231l socket2124 210 44795-44816
START_RANGE client231a socket2125 210 44816-44837
START_RANGE client231b socket2126 210 44837-44858
START_RANGE client231c socket2127 210 44858-44879
START_RANGE client231d socket2128 210 44879-44900
START_RANGE client231e socket2129 210 44900-44921
START_RANGE client231f socket2130 220 44921-44943
START_RANGE client231g socket2131 210 44943-44964
START_RANGE client231h socket2132 210 44964-44985
START_RANGE client231i socket2133 210 44985-45006
START_RANGE client231j socket2134 210 45006-45027
START_RANGE client231k socket2135 210 45027-45048
START_RANGE client231l socket2136 210 45048-45069
START_RANGE client231a socket2137 210 45069-45090
START_RANGE client231b socket2138 210 45090-45111
START_RANGE client231c socket2139 210 45111-45132
START_RANGE client231d socket2140 220 45132-45154
START_RANGE client231e socket2141 210 45154-45175
START_RANGE client231f socket2142 210 45175-45196
START_RANGE client231g socket2143 210 45196-45217
START_RANGE client231h socket2144 210 45217-45238
START_RANGE client231i socket2145 210 45238-45259
START_RANGE client231j socket2146 210 45259-45280
START_RANGE client231k socket2147 210 45280-45301
START_RANGE client231l socket2148 210 45301-45322
START_RANGE client231a socket2149 210 45322-45343
START_RANGE client231b socket2150 220 45343-45365
START_RANGE client231c socket2151 210 45365-45386
START_RANGE client231d socket2152 210 45386-45407
START_RANGE client231e socket2153 210 45407-45428
START_RANGE client231f socket2154 210 45428-45449
START_RANGE client231g socket2155 210 45449-45470
START_RANGE client231h socket2156 210 45470-45491
START_RANGE client231i socket2157 210 45491-45512
START_RANGE client231j socket2158 210 45512-45533
START_RANGE client231k socket2159 210 45533-45554
START_RANGE client231l socket2160 220 45554-45576

START_RANGE client232a socket2161 210 45576-45597
START_RANGE client232b socket2162 210 45597-45618
START_RANGE client232c socket2163 210 45618-45639
START_RANGE client232d socket2164 210 45639-45660
START_RANGE client232e socket2165 210 45660-45681
START_RANGE client232f socket2166 210 45681-45702
START_RANGE client232g socket2167 210 45702-45723
START_RANGE client232h socket2168 210 45723-45744
START_RANGE client232i socket2169 210 45744-45765
START_RANGE client232j socket2170 220 45765-45787
START_RANGE client232k socket2171 210 45787-45808
START_RANGE client232l socket2172 210 45808-45829
START_RANGE client232a socket2173 210 45829-45850
START_RANGE client232b socket2174 210 45850-45871
START_RANGE client232c socket2175 210 45871-45892
START_RANGE client232d socket2176 210 45892-45913
START_RANGE client232e socket2177 210 45913-45934
START_RANGE client232f socket2178 210 45934-45955
START_RANGE client232g socket2179 210 45955-45976
START_RANGE client232h socket2180 220 45976-45998
START_RANGE client232i socket2181 210 45998-46019
START_RANGE client232j socket2182 210 46019-46040
START_RANGE client232k socket2183 210 46040-46061
START_RANGE client232l socket2184 210 46061-46082
START_RANGE client232a socket2185 210 46082-46103
START_RANGE client232b socket2186 210 46103-46124
START_RANGE client232c socket2187 210 46124-46145
START_RANGE client232d socket2188 210 46145-46166
START_RANGE client232e socket2189 210 46166-46187
START_RANGE client232f socket2190 220 46187-46209
START_RANGE client232g socket2191 210 46209-46230
START_RANGE client232h socket2192 210 46230-46251
START_RANGE client232i socket2193 210 46251-46272
START_RANGE client232j socket2194 210 46272-46293
START_RANGE client232k socket2195 210 46293-46314
START_RANGE client232l socket2196 210 46314-46335
START_RANGE client232a socket2197 210 46335-46356
START_RANGE client232b socket2198 210 46356-46377
START_RANGE client232c socket2199 210 46377-46398
START_RANGE client232d socket2200 220 46398-46420
START_RANGE client232e socket2201 210 46420-46441
START_RANGE client232f socket2202 210 46441-46462
START_RANGE client232g socket2203 210 46462-46483
START_RANGE client232h socket2204 210 46483-46504
START_RANGE client232i socket2205 210 46504-46525
START_RANGE client232j socket2206 210 46525-46546
START_RANGE client232k socket2207 210 46546-46567
START_RANGE client232l socket2208 210 46567-46588

START_RANGE client241a socket2209 210 46588-46609
START_RANGE client241b socket2210 220 46609-46631
START_RANGE client241c socket2211 210 46631-46652
START_RANGE client241d socket2212 210 46652-46673
START_RANGE client241e socket2213 210 46673-46694
START_RANGE client241f socket2214 210 46694-46715
START_RANGE client241g socket2215 210 46715-46736
START_RANGE client241h socket2216 210 46736-46757
START_RANGE client241i socket2217 210 46757-46778
START_RANGE client241j socket2218 210 46778-46799
START_RANGE client241k socket2219 210 46799-46820
START_RANGE client241l socket2220 220 46820-46842
START_RANGE client241a socket2221 210 46842-46863
START_RANGE client241b socket2222 210 46863-46884
START_RANGE client241c socket2223 210 46884-46905
START_RANGE client241d socket2224 210 46905-46926
START_RANGE client241e socket2225 210 46926-46947
START_RANGE client241f socket2226 210 46947-46968
START_RANGE client241g socket2227 210 46968-46989
START_RANGE client241h socket2228 210 46989-47010
START_RANGE client241i socket2229 210 47010-47031
START_RANGE client241j socket2230 220 47031-47053
START_RANGE client241k socket2231 210 47053-47074
START_RANGE client241l socket2232 210 47074-47095
START_RANGE client241a socket2233 210 47095-47116
START_RANGE client241b socket2234 210 47116-47137
START_RANGE client241c socket2235 210 47137-47158
START_RANGE client241d socket2236 210 47158-47179
START_RANGE client241e socket2237 210 47179-47200
START_RANGE client241f socket2238 210 47200-47221
START_RANGE client241g socket2239 210 47221-47242
START_RANGE client241h socket2240 220 47242-47264
START_RANGE client241i socket2241 210 47264-47285
START_RANGE client241j socket2242 210 47285-47306
START_RANGE client241k socket2243 210 47306-47327
START_RANGE client241l socket2244 210 47327-47348
START_RANGE client241a socket2245 210 47348-47369
START_RANGE client241b socket2246 210 47369-47390
START_RANGE client241c socket2247 210 47390-47411
START_RANGE client241d socket2248 210 47411-47432

START_RANGE client241e socket2249 210 47432-47453
START_RANGE client241f socket2250 220 47453-47475
START_RANGE client241g socket2251 210 47475-47496
START_RANGE client241h socket2252 210 47496-47517
START_RANGE client241i socket2253 210 47517-47538
START_RANGE client241j socket2254 210 47538-47559
START_RANGE client241k socket2255 210 47559-47580
START_RANGE client241l socket2256 210 47580-47601

START_RANGE client242a socket2257 210 47601-47622
START_RANGE client242b socket2258 210 47622-47643
START_RANGE client242c socket2259 210 47643-47664
START_RANGE client242d socket2260 220 47664-47686
START_RANGE client242e socket2261 210 47686-47707
START_RANGE client242f socket2262 210 47707-47728
START_RANGE client242g socket2263 210 47728-47749
START_RANGE client242h socket2264 210 47749-47770
START_RANGE client242i socket2265 210 47770-47791
START_RANGE client242j socket2266 210 47791-47812
START_RANGE client242k socket2267 210 47812-47833
START_RANGE client242l socket2268 210 47833-47854
START_RANGE client242a socket2269 210 47854-47875
START_RANGE client242b socket2270 220 47875-47897
START_RANGE client242c socket2271 210 47897-47918
START_RANGE client242d socket2272 210 47918-47939
START_RANGE client242e socket2273 210 47939-47960
START_RANGE client242f socket2274 210 47960-47981
START_RANGE client242g socket2275 210 47981-48002
START_RANGE client242h socket2276 210 48002-48023
START_RANGE client242i socket2277 210 48023-48044
START_RANGE client242j socket2278 210 48044-48065
START_RANGE client242k socket2279 210 48065-48086
START_RANGE client242l socket2280 220 48086-48108
START_RANGE client242a socket2281 210 48108-48129
START_RANGE client242b socket2282 210 48129-48150
START_RANGE client242c socket2283 210 48150-48171
START_RANGE client242d socket2284 210 48171-48192
START_RANGE client242e socket2285 210 48192-48213
START_RANGE client242f socket2286 210 48213-48234
START_RANGE client242g socket2287 210 48234-48255
START_RANGE client242h socket2288 210 48255-48276
START_RANGE client242i socket2289 210 48276-48297
START_RANGE client242j socket2290 220 48297-48319
START_RANGE client242k socket2291 210 48319-48340
START_RANGE client242l socket2292 210 48340-48361
START_RANGE client242a socket2293 210 48361-48382
START_RANGE client242b socket2294 210 48382-48403
START_RANGE client242c socket2295 210 48403-48424
START_RANGE client242d socket2296 210 48424-48445
START_RANGE client242e socket2297 210 48445-48466
START_RANGE client242f socket2298 210 48466-48487
START_RANGE client242g socket2299 210 48487-48508
START_RANGE client242h socket2300 220 48508-48530
START_RANGE client242i socket2301 210 48530-48551
START_RANGE client242j socket2302 210 48551-48572
START_RANGE client242k socket2303 210 48572-48593
START_RANGE client242l socket2304 210 48593-48614

#elif MASTER_NUM9
START_RANGE client251a socket2305 210 48614-48635
START_RANGE client251b socket2306 210 48635-48656
START_RANGE client251c socket2307 210 48656-48677
START_RANGE client251d socket2308 210 48677-48698
START_RANGE client251e socket2309 210 48698-48719
START_RANGE client251f socket2310 220 48719-48741
START_RANGE client251g socket2311 210 48741-48762
START_RANGE client251h socket2312 210 48762-48783
START_RANGE client251i socket2313 210 48783-48804
START_RANGE client251j socket2314 210 48804-48825
START_RANGE client251k socket2315 210 48825-48846
START_RANGE client251l socket2316 210 48846-48867
START_RANGE client251a socket2317 210 48867-48888
START_RANGE client251b socket2318 210 48888-48909
START_RANGE client251c socket2319 210 48909-48930
START_RANGE client251d socket2320 220 48930-48952
START_RANGE client251e socket2321 210 48952-48973
START_RANGE client251f socket2322 210 48973-48994
START_RANGE client251g socket2323 210 48994-49015
START_RANGE client251h socket2324 210 49015-49036
START_RANGE client251i socket2325 210 49036-49057
START_RANGE client251j socket2326 210 49057-49078
START_RANGE client251k socket2327 210 49078-49099
START_RANGE client251l socket2328 210 49099-49120
START_RANGE client251a socket2329 210 49120-49141
START_RANGE client251b socket2330 220 49141-49163
START_RANGE client251c socket2331 210 49163-49184
START_RANGE client251d socket2332 210 49184-49205
START_RANGE client251e socket2333 210 49205-49226
START_RANGE client251f socket2334 210 49226-49247
START_RANGE client251g socket2335 210 49247-49268
START_RANGE client251h socket2336 210 49268-49289
START_RANGE client251i socket2337 210 49289-49310
START_RANGE client251j socket2338 210 49310-49331
START_RANGE client251k socket2339 210 49331-49352
START_RANGE client251l socket2340 220 49352-49374
START_RANGE client251a socket2341 210 49374-49395
START_RANGE client251b socket2342 210 49395-49416
START_RANGE client251c socket2343 210 49416-49437
START_RANGE client251d socket2344 210 49437-49458
START_RANGE client251e socket2345 210 49458-49479
START_RANGE client251f socket2346 210 49479-49500
START_RANGE client251g socket2347 210 49500-49521
START_RANGE client251h socket2348 210 49521-49542
START_RANGE client251i socket2349 210 49542-49563
START_RANGE client251j socket2350 220 49563-49585
START_RANGE client251k socket2351 210 49585-49606
START_RANGE client251l socket2352 210 49606-49627

START_RANGE client252a socket2353 210 49627-49648
START_RANGE client252b socket2354 210 49648-49669
START_RANGE client252c socket2355 210 49669-49690
START_RANGE client252d socket2356 210 49690-49711
START_RANGE client252e socket2357 210 49711-49732
START_RANGE client252f socket2358 210 49732-49753
START_RANGE client252g socket2359 210 49753-49774
START_RANGE client252h socket2360 220 49774-49796
START_RANGE client252i socket2361 210 49796-49817
START_RANGE client252j socket2362 210 49817-49838
START_RANGE client252k socket2363 210 49838-49859
START_RANGE client252l socket2364 210 49859-49880
START_RANGE client252a socket2365 210 49880-49901
START_RANGE client252b socket2366 210 49901-49922
START_RANGE client252c socket2367 210 49922-49943
START_RANGE client252d socket2368 210 49943-49964
START_RANGE client252e socket2369 210 49964-49985
START_RANGE client252f socket2370 220 49985-50007
START_RANGE client252g socket2371 210 50007-50028
START_RANGE client252h socket2372 210 50028-50049
START_RANGE client252i socket2373 210 50049-50070
START_RANGE client252j socket2374 210 50070-50091
START_RANGE client252k socket2375 210 50091-50112
START_RANGE client252l socket2376 210 50112-50133
START_RANGE client252a socket2377 210 50133-50154
START_RANGE client252b socket2378 210 50154-50175
START_RANGE client252c socket2379 210 50175-50196
START_RANGE client252d socket2380 220 50196-50218
START_RANGE client252e socket2381 210 50218-50239
START_RANGE client252f socket2382 210 50239-50260
START_RANGE client252g socket2383 210 50260-50281
START_RANGE client252h socket2384 210 50281-50302
START_RANGE client252i socket2385 210 50302-50323
START_RANGE client252j socket2386 210 50323-50344
START_RANGE client252k socket2387 210 50344-50365
START_RANGE client252l socket2388 210 50365-50386
START_RANGE client252a socket2389 210 50386-50407
START_RANGE client252b socket2390 220 50407-50429
START_RANGE client252c socket2391 210 50429-50450
START_RANGE client252d socket2392 210 50450-50471
START_RANGE client252e socket2393 210 50471-50492
START_RANGE client252f socket2394 210 50492-50513
START_RANGE client252g socket2395 210 50513-50534
START_RANGE client252h socket2396 210 50534-50555
START_RANGE client252i socket2397 210 50555-50576
START_RANGE client252j socket2398 210 50576-50597
START_RANGE client252k socket2399 210 50597-50618
START_RANGE client252l socket2400 220 50618-50640

START_RANGE client261a socket2401 210 50640-50661
START_RANGE client261b socket2402 210 50661-50682
START_RANGE client261c socket2403 210 50682-50703
START_RANGE client261d socket2404 210 50703-50724
START_RANGE client261e socket2405 210 50724-50745
START_RANGE client261f socket2406 210 50745-50766
START_RANGE client261g socket2407 210 50766-50787
START_RANGE client261h socket2408 210 50787-50808
START_RANGE client261i socket2409 210 50808-50829
START_RANGE client261j socket2410 220 50829-50851
START_RANGE client261k socket2411 210 50851-50872
START_RANGE client261l socket2412 210 50872-50893
START_RANGE client261a socket2413 210 50893-50914
START_RANGE client261b socket2414 210 50914-50935
START_RANGE client261c socket2415 210 50935-50956
START_RANGE client261d socket2416 210 50956-50977
START_RANGE client261e socket2417 210 50977-50998
START_RANGE client261f socket2418 210 50998-51019
START_RANGE client261g socket2419 210 51019-51040
START_RANGE client261h socket2420 220 51040-51062
START_RANGE client261i socket2421 210 51062-51083
START_RANGE client261j socket2422 210 51083-51104
START_RANGE client261k socket2423 210 51104-51125
START_RANGE client261l socket2424 210 51125-51146
START_RANGE client261a socket2425 210 51146-51167
START_RANGE client261b socket2426 210 51167-51188
START_RANGE client261c socket2427 210 51188-51209
START_RANGE client261d socket2428 210 51209-51230
START_RANGE client261e socket2429 210 51230-51251
START_RANGE client261f socket2430 220 51251-51273
START_RANGE client261g socket2431 210 51273-51294
START_RANGE client261h socket2432 210 51294-51315
START_RANGE client261i socket2433 210 51315-51336
START_RANGE client261j socket2434 210 51336-51357
START_RANGE client261k socket2435 210 51357-51378
START_RANGE client261l socket2436 210 51378-51399
START_RANGE client261a socket2437 210 51399-51420
START_RANGE client261b socket2438 210 51420-51441
START_RANGE client261c socket2439 210 51441-51462
START_RANGE client261d socket2440 220 51462-51484
START_RANGE client261e socket2441 210 51484-51505

START_RANGE client261f socket2442 210 51505-51526
START_RANGE client261g socket2443 210 51526-51547
START_RANGE client261h socket2444 210 51547-51568
START_RANGE client261i socket2445 210 51568-51589
START_RANGE client261j socket2446 210 51589-51610
START_RANGE client261k socket2447 210 51610-51631
START_RANGE client261l socket2448 210 51631-51652

START_RANGE client262a socket2449 210 51652-51673
START_RANGE client262b socket2450 220 51673-51695
START_RANGE client262c socket2451 210 51695-51716
START_RANGE client262d socket2452 210 51716-51737
START_RANGE client262e socket2453 210 51737-51758
START_RANGE client262f socket2454 210 51758-51779
START_RANGE client262g socket2455 210 51779-51800
START_RANGE client262h socket2456 210 51800-51821
START_RANGE client262i socket2457 210 51821-51842
START_RANGE client262j socket2458 210 51842-51863
START_RANGE client262k socket2459 210 51863-51884
START_RANGE client262l socket2460 220 51884-51906
START_RANGE client262a socket2461 210 51906-51927
START_RANGE client262b socket2462 210 51927-51948
START_RANGE client262c socket2463 210 51948-51969
START_RANGE client262d socket2464 210 51969-51990
START_RANGE client262e socket2465 210 51990-52011
START_RANGE client262f socket2466 210 52011-52032
START_RANGE client262g socket2467 210 52032-52053
START_RANGE client262h socket2468 210 52053-52074
START_RANGE client262i socket2469 210 52074-52095
START_RANGE client262j socket2470 220 52095-52117
START_RANGE client262k socket2471 210 52117-52138
START_RANGE client262l socket2472 210 52138-52159
START_RANGE client262a socket2473 210 52159-52180
START_RANGE client262b socket2474 210 52180-52201
START_RANGE client262c socket2475 210 52201-52222
START_RANGE client262d socket2476 210 52222-52243
START_RANGE client262e socket2477 210 52243-52264
START_RANGE client262f socket2478 210 52264-52285
START_RANGE client262g socket2479 210 52285-52306
START_RANGE client262h socket2480 220 52306-52328
START_RANGE client262i socket2481 210 52328-52349
START_RANGE client262j socket2482 210 52349-52370
START_RANGE client262k socket2483 210 52370-52391
START_RANGE client262l socket2484 210 52391-52412
START_RANGE client262a socket2485 210 52412-52433
START_RANGE client262b socket2486 210 52433-52454
START_RANGE client262c socket2487 210 52454-52475
START_RANGE client262d socket2488 210 52475-52496
START_RANGE client262e socket2489 210 52496-52517
START_RANGE client262f socket2490 220 52517-52539
START_RANGE client262g socket2491 210 52539-52560
START_RANGE client262h socket2492 210 52560-52581
START_RANGE client262i socket2493 210 52581-52602
START_RANGE client262j socket2494 210 52602-52623
START_RANGE client262k socket2495 210 52623-52644
START_RANGE client262l socket2496 210 52644-52665

START_RANGE client271a socket2497 210 52665-52686
START_RANGE client271b socket2498 210 52686-52707
START_RANGE client271c socket2499 210 52707-52728
START_RANGE client271d socket2500 220 52728-52750
START_RANGE client271e socket2501 210 52750-52771
START_RANGE client271f socket2502 210 52771-52792
START_RANGE client271g socket2503 210 52792-52813
START_RANGE client271h socket2504 210 52813-52834
START_RANGE client271i socket2505 210 52834-52855
START_RANGE client271j socket2506 210 52855-52876
START_RANGE client271k socket2507 210 52876-52897
START_RANGE client271l socket2508 210 52897-52918
START_RANGE client271a socket2509 210 52918-52939
START_RANGE client271b socket2510 220 52939-52961
START_RANGE client271c socket2511 210 52961-52982
START_RANGE client271d socket2512 210 52982-53003
START_RANGE client271e socket2513 210 53003-53024
START_RANGE client271f socket2514 210 53024-53045
START_RANGE client271g socket2515 210 53045-53066
START_RANGE client271h socket2516 210 53066-53087
START_RANGE client271i socket2517 210 53087-53108
START_RANGE client271j socket2518 210 53108-53129
START_RANGE client271k socket2519 210 53129-53150
START_RANGE client271l socket2520 220 53150-53172
START_RANGE client271a socket2521 210 53172-53193
START_RANGE client271b socket2522 210 53193-53214
START_RANGE client271c socket2523 210 53214-53235
START_RANGE client271d socket2524 210 53235-53256
START_RANGE client271e socket2525 210 53256-53277
START_RANGE client271f socket2526 210 53277-53298
START_RANGE client271g socket2527 210 53298-53319
START_RANGE client271h socket2528 210 53319-53340
START_RANGE client271i socket2529 210 53340-53361
START_RANGE client271j socket2530 220 53361-53383
START_RANGE client271k socket2531 210 53383-53404
START_RANGE client271l socket2532 210 53404-53425
START_RANGE client271a socket2533 210 53425-53446
START_RANGE client271b socket2534 210 53446-53467
START_RANGE client271c socket2535 210 53467-53488
START_RANGE client271d socket2536 210 53488-53509
START_RANGE client271e socket2537 210 53509-53530
START_RANGE client271f socket2538 210 53530-53551
START_RANGE client271g socket2539 210 53551-53572
START_RANGE client271h socket2540 220 53572-53594
START_RANGE client271i socket2541 210 53594-53615
START_RANGE client271j socket2542 210 53615-53636
START_RANGE client271k socket2543 210 53636-53657
START_RANGE client271l socket2544 210 53657-53678

START_RANGE client272a socket2545 210 53678-53699
START_RANGE client272b socket2546 210 53699-53720
START_RANGE client272c socket2547 210 53720-53741
START_RANGE client272d socket2548 210 53741-53762
START_RANGE client272e socket2549 210 53762-53783
START_RANGE client272f socket2550 220 53783-53805
START_RANGE client272g socket2551 210 53805-53826
START_RANGE client272h socket2552 210 53826-53847
START_RANGE client272i socket2553 210 53847-53868
START_RANGE client272j socket2554 210 53868-53889
START_RANGE client272k socket2555 210 53889-53910
START_RANGE client272l socket2556 210 53910-53931
START_RANGE client272a socket2557 210 53931-53952
START_RANGE client272b socket2558 210 53952-53973
START_RANGE client272c socket2559 210 53973-53994
START_RANGE client272d socket2560 220 53994-54016
START_RANGE client272e socket2561 210 54016-54037
START_RANGE client272f socket2562 210 54037-54058
START_RANGE client272g socket2563 210 54058-54079
START_RANGE client272h socket2564 210 54079-54100
START_RANGE client272i socket2565 210 54100-54121
START_RANGE client272j socket2566 210 54121-54142
START_RANGE client272k socket2567 210 54142-54163
START_RANGE client272l socket2568 210 54163-54184
START_RANGE client272a socket2569 210 54184-54205
START_RANGE client272b socket2570 220 54205-54227
START_RANGE client272c socket2571 210 54227-54248
START_RANGE client272d socket2572 210 54248-54269
START_RANGE client272e socket2573 210 54269-54290
START_RANGE client272f socket2574 210 54290-54311
START_RANGE client272g socket2575 210 54311-54332
START_RANGE client272h socket2576 210 54332-54353
START_RANGE client272i socket2577 210 54353-54374
START_RANGE client272j socket2578 210 54374-54395
START_RANGE client272k socket2579 210 54395-54416
START_RANGE client272l socket2580 220 54416-54438
START_RANGE client272a socket2581 210 54438-54459
START_RANGE client272b socket2582 210 54459-54480
START_RANGE client272c socket2583 210 54480-54501
START_RANGE client272d socket2584 210 54501-54522
START_RANGE client272e socket2585 210 54522-54543
START_RANGE client272f socket2586 210 54543-54564
START_RANGE client272g socket2587 210 54564-54585
START_RANGE client272h socket2588 210 54585-54606
START_RANGE client272i socket2589 210 54606-54627
START_RANGE client272j socket2590 220 54627-54649
START_RANGE client272k socket2591 210 54649-54670
START_RANGE client272l socket2592 210 54670-54691

#elif MASTER_NUM10
START_RANGE client281a socket2593 210 54691-54712
START_RANGE client281b socket2594 210 54712-54733
START_RANGE client281c socket2595 210 54733-54754
START_RANGE client281d socket2596 210 54754-54775
START_RANGE client281e socket2597 210 54775-54796
START_RANGE client281f socket2598 210 54796-54817
START_RANGE client281g socket2599 210 54817-54838
START_RANGE client281h socket2600 220 54838-54860
START_RANGE client281i socket2601 210 54860-54881
START_RANGE client281j socket2602 210 54881-54902
START_RANGE client281k socket2603 210 54902-54923
START_RANGE client281l socket2604 210 54923-54944
START_RANGE client281a socket2605 210 54944-54965
START_RANGE client281b socket2606 210 54965-54986
START_RANGE client281c socket2607 210 54986-55007
START_RANGE client281d socket2608 210 55007-55028
START_RANGE client281e socket2609 210 55028-55049
START_RANGE client281f socket2610 220 55049-55071
START_RANGE client281g socket2611 210 55071-55092
START_RANGE client281h socket2612 210 55092-55113
START_RANGE client281i socket2613 210 55113-55134
START_RANGE client281j socket2614 210 55134-55155
START_RANGE client281k socket2615 210 55155-55176
START_RANGE client281l socket2616 210 55176-55197
START_RANGE client281a socket2617 210 55197-55218
START_RANGE client281b socket2618 210 55218-55239
START_RANGE client281c socket2619 210 55239-55260
START_RANGE client281d socket2620 220 55260-55282
START_RANGE client281e socket2621 210 55282-55303
START_RANGE client281f socket2622 210 55303-55324
START_RANGE client281g socket2623 210 55324-55345
START_RANGE client281h socket2624 210 55345-55366
START_RANGE client281i socket2625 210 55366-55387
START_RANGE client281j socket2626 210 55387-55408
START_RANGE client281k socket2627 210 55408-55429
START_RANGE client281l socket2628 210 55429-55450
START_RANGE client281a socket2629 210 55450-55471
START_RANGE client281b socket2630 220 55471-55493
START_RANGE client281c socket2631 210 55493-55514
START_RANGE client281d socket2632 210 55514-55535
START_RANGE client281e socket2633 210 55535-55556
START_RANGE client281f socket2634 210 55556-55577

START_RANGE client281g socket2635 210 55577-55598
START_RANGE client281h socket2636 210 55598-55619
START_RANGE client281i socket2637 210 55619-55640
START_RANGE client281j socket2638 210 55640-55661
START_RANGE client281k socket2639 210 55661-55682
START_RANGE client281l socket2640 220 55682-55704

START_RANGE client282a socket2641 210 55704-55725
START_RANGE client282b socket2642 210 55725-55746
START_RANGE client282c socket2643 210 55746-55767
START_RANGE client282d socket2644 210 55767-55788
START_RANGE client282e socket2645 210 55788-55809
START_RANGE client282f socket2646 210 55809-55830
START_RANGE client282g socket2647 210 55830-55851
START_RANGE client282h socket2648 210 55851-55872
START_RANGE client282i socket2649 210 55872-55893
START_RANGE client282j socket2650 220 55893-55915
START_RANGE client282k socket2651 210 55915-55936
START_RANGE client282l socket2652 210 55936-55957
START_RANGE client282a socket2653 210 55957-55978
START_RANGE client282b socket2654 210 55978-55999
START_RANGE client282c socket2655 210 55999-56020
START_RANGE client282d socket2656 210 56020-56041
START_RANGE client282e socket2657 210 56041-56062
START_RANGE client282f socket2658 210 56062-56083
START_RANGE client282g socket2659 210 56083-56104
START_RANGE client282h socket2660 220 56104-56126
START_RANGE client282i socket2661 210 56126-56147
START_RANGE client282j socket2662 210 56147-56168
START_RANGE client282k socket2663 210 56168-56189
START_RANGE client282l socket2664 210 56189-56210
START_RANGE client282a socket2665 210 56210-56231
START_RANGE client282b socket2666 210 56231-56252
START_RANGE client282c socket2667 210 56252-56273
START_RANGE client282d socket2668 210 56273-56294
START_RANGE client282e socket2669 210 56294-56315
START_RANGE client282f socket2670 220 56315-56337
START_RANGE client282g socket2671 210 56337-56358
START_RANGE client282h socket2672 210 56358-56379
START_RANGE client282i socket2673 210 56379-56400
START_RANGE client282j socket2674 210 56400-56421
START_RANGE client282k socket2675 210 56421-56442
START_RANGE client282l socket2676 210 56442-56463
START_RANGE client282a socket2677 210 56463-56484
START_RANGE client282b socket2678 210 56484-56505
START_RANGE client282c socket2679 210 56505-56526
START_RANGE client282d socket2680 220 56526-56548
START_RANGE client282e socket2681 210 56548-56569
START_RANGE client282f socket2682 210 56569-56590
START_RANGE client282g socket2683 210 56590-56611
START_RANGE client282h socket2684 210 56611-56632
START_RANGE client282i socket2685 210 56632-56653
START_RANGE client282j socket2686 210 56653-56674
START_RANGE client282k socket2687 210 56674-56695
START_RANGE client282l socket2688 210 56695-56716

START_RANGE client291a socket2689 210 56716-56737
START_RANGE client291b socket2690 220 56737-56759
START_RANGE client291c socket2691 210 56759-56780
START_RANGE client291d socket2692 210 56780-56801
START_RANGE client291e socket2693 210 56801-56822
START_RANGE client291f socket2694 210 56822-56843
START_RANGE client291g socket2695 210 56843-56864
START_RANGE client291h socket2696 210 56864-56885
START_RANGE client291i socket2697 210 56885-56906
START_RANGE client291j socket2698 210 56906-56927
START_RANGE client291k socket2699 210 56927-56948
START_RANGE client291l socket2700 220 56948-56970
START_RANGE client291a socket2701 210 56970-56991
START_RANGE client291b socket2702 210 56991-57012
START_RANGE client291c socket2703 210 57012-57033
START_RANGE client291d socket2704 210 57033-57054
START_RANGE client291e socket2705 210 57054-57075
START_RANGE client291f socket2706 210 57075-57096
START_RANGE client291g socket2707 210 57096-57117
START_RANGE client291h socket2708 210 57117-57138
START_RANGE client291i socket2709 210 57138-57159
START_RANGE client291j socket2710 220 57159-57181
START_RANGE client291k socket2711 210 57181-57202
START_RANGE client291l socket2712 210 57202-57223
START_RANGE client291a socket2713 210 57223-57244
START_RANGE client291b socket2714 210 57244-57265
START_RANGE client291c socket2715 210 57265-57286
START_RANGE client291d socket2716 210 57286-57307
START_RANGE client291e socket2717 210 57307-57328
START_RANGE client291f socket2718 210 57328-57349
START_RANGE client291g socket2719 210 57349-57370
START_RANGE client291h socket2720 220 57370-57392
START_RANGE client291i socket2721 210 57392-57413
START_RANGE client291j socket2722 210 57413-57434
START_RANGE client291k socket2723 210 57434-57455
START_RANGE client291l socket2724 210 57455-57476
START_RANGE client291a socket2725 210 57476-57497
START_RANGE client291b socket2726 210 57497-57518
START_RANGE client291c socket2727 210 57518-57539
START_RANGE client291d socket2728 210 57539-57560
START_RANGE client291e socket2729 210 57560-57581
START_RANGE client291f socket2730 220 57581-57603
START_RANGE client291g socket2731 210 57603-57624
START_RANGE client291h socket2732 210 57624-57645
START_RANGE client291i socket2733 210 57645-57666
START_RANGE client291j socket2734 210 57666-57687
START_RANGE client291k socket2735 210 57687-57708
START_RANGE client291l socket2736 210 57708-57729

START_RANGE client292a socket2737 210 57729-57750
START_RANGE client292b socket2738 210 57750-57771
START_RANGE client292c socket2739 210 57771-57792
START_RANGE client292d socket2740 220 57792-57814
START_RANGE client292e socket2741 210 57814-57835
START_RANGE client292f socket2742 210 57835-57856
START_RANGE client292g socket2743 210 57856-57877
START_RANGE client292h socket2744 210 57877-57898
START_RANGE client292i socket2745 210 57898-57919
START_RANGE client292j socket2746 210 57919-57940
START_RANGE client292k socket2747 210 57940-57961
START_RANGE client292l socket2748 210 57961-57982
START_RANGE client292a socket2749 210 57982-58003
START_RANGE client292b socket2750 220 58003-58025
START_RANGE client292c socket2751 210 58025-58046
START_RANGE client292d socket2752 210 58046-58067
START_RANGE client292e socket2753 210 58067-58088
START_RANGE client292f socket2754 210 58088-58109
START_RANGE client292g socket2755 210 58109-58130
START_RANGE client292h socket2756 210 58130-58151
START_RANGE client292i socket2757 210 58151-58172
START_RANGE client292j socket2758 210 58172-58193
START_RANGE client292k socket2759 210 58193-58214
START_RANGE client292l socket2760 220 58214-58236
START_RANGE client292a socket2761 210 58236-58257
START_RANGE client292b socket2762 210 58257-58278
START_RANGE client292c socket2763 210 58278-58299
START_RANGE client292d socket2764 210 58299-58320
START_RANGE client292e socket2765 210 58320-58341
START_RANGE client292f socket2766 210 58341-58362
START_RANGE client292g socket2767 210 58362-58383
START_RANGE client292h socket2768 210 58383-58404
START_RANGE client292i socket2769 210 58404-58425
START_RANGE client292j socket2770 220 58425-58447
START_RANGE client292k socket2771 210 58447-58468
START_RANGE client292l socket2772 210 58468-58489
START_RANGE client292a socket2773 210 58489-58510
START_RANGE client292b socket2774 210 58510-58531
START_RANGE client292c socket2775 210 58531-58552
START_RANGE client292d socket2776 210 58552-58573
START_RANGE client292e socket2777 210 58573-58594
START_RANGE client292f socket2778 210 58594-58615
START_RANGE client292g socket2779 210 58615-58636
START_RANGE client292h socket2780 220 58636-58658
START_RANGE client292i socket2781 210 58658-58679
START_RANGE client292j socket2782 210 58679-58700
START_RANGE client292k socket2783 210 58700-58721
START_RANGE client292l socket2784 210 58721-58742

START_RANGE client301a socket2785 210 58742-58763
START_RANGE client301b socket2786 210 58763-58784
START_RANGE client301c socket2787 210 58784-58805
START_RANGE client301d socket2788 210 58805-58826
START_RANGE client301e socket2789 210 58826-58847
START_RANGE client301f socket2790 220 58847-58869
START_RANGE client301g socket2791 210 58869-58890
START_RANGE client301h socket2792 210 58890-58911
START_RANGE client301i socket2793 210 58911-58932
START_RANGE client301j socket2794 210 58932-58953
START_RANGE client301k socket2795 210 58953-58974
START_RANGE client301l socket2796 210 58974-58995
START_RANGE client301a socket2797 210 58995-59016
START_RANGE client301b socket2798 210 59016-59037
START_RANGE client301c socket2799 210 59037-59058
START_RANGE client301d socket2800 220 59058-59080
START_RANGE client301e socket2801 210 59080-59101
START_RANGE client301f socket2802 210 59101-59122
START_RANGE client301g socket2803 210 59122-59143
START_RANGE client301h socket2804 210 59143-59164
START_RANGE client301i socket2805 210 59164-59185
START_RANGE client301j socket2806 210 59185-59206
START_RANGE client301k socket2807 210 59206-59227
START_RANGE client301l socket2808 210 59227-59248
START_RANGE client301a socket2809 210 59248-59269
START_RANGE client301b socket2810 220 59269-59291
START_RANGE client301c socket2811 210 59291-59312
START_RANGE client301d socket2812 210 59312-59333
START_RANGE client301e socket2813 210 59333-59354
START_RANGE client301f socket2814 210 59354-59375
START_RANGE client301g socket2815 210 59375-59396
START_RANGE client301h socket2816 210 59396-59417
START_RANGE client301i socket2817 210 59417-59438
START_RANGE client301j socket2818 210 59438-59459
START_RANGE client301k socket2819 210 59459-59480
START_RANGE client301l socket2820 220 59480-59502
START_RANGE client301a socket2821 210 59502-59523
START_RANGE client301b socket2822 210 59523-59544
START_RANGE client301c socket2823 210 59544-59565
START_RANGE client301d socket2824 210 59565-59586
START_RANGE client301e socket2825 210 59586-59607
START_RANGE client301f socket2826 210 59607-59628
START_RANGE client301g socket2827 210 59628-59649
START_RANGE client301h socket2828 210 59649-59670

```
START_RANGE client301i socket2829 210 59670-59691
START_RANGE client301j socket2830 220 59691-59713
START_RANGE client301k socket2831 210 59713-59734
START_RANGE client301l socket2832 210 59734-59755

START_RANGE client302a socket2833 210 59755-59776
START_RANGE client302b socket2834 210 59776-59797
START_RANGE client302c socket2835 210 59797-59818
START_RANGE client302d socket2836 210 59818-59839
START_RANGE client302e socket2837 210 59839-59860
START_RANGE client302f socket2838 210 59860-59881
START_RANGE client302g socket2839 210 59881-59902
START_RANGE client302h socket2840 220 59902-59924
START_RANGE client302i socket2841 210 59924-59945
START_RANGE client302j socket2842 210 59945-59966
START_RANGE client302k socket2843 210 59966-59987
START_RANGE client302l socket2844 210 59987-60008
START_RANGE client302a socket2845 210 60008-60029
START_RANGE client302b socket2846 210 60029-60050
START_RANGE client302c socket2847 210 60050-60071
START_RANGE client302d socket2848 210 60071-60092
START_RANGE client302e socket2849 210 60092-60113
START_RANGE client302f socket2850 220 60113-60135
START_RANGE client302g socket2851 210 60135-60156
START_RANGE client302h socket2852 210 60156-60177
START_RANGE client302i socket2853 210 60177-60198
START_RANGE client302j socket2854 210 60198-60219
START_RANGE client302k socket2855 210 60219-60240
START_RANGE client302l socket2856 210 60240-60261
START_RANGE client302a socket2857 210 60261-60282
START_RANGE client302b socket2858 210 60282-60303
START_RANGE client302c socket2859 210 60303-60324
START_RANGE client302d socket2860 220 60324-60346
START_RANGE client302e socket2861 210 60346-60367
START_RANGE client302f socket2862 210 60367-60388
START_RANGE client302g socket2863 210 60388-60409
START_RANGE client302h socket2864 210 60409-60430
START_RANGE client302i socket2865 210 60430-60451
START_RANGE client302j socket2866 210 60451-60472
START_RANGE client302k socket2867 210 60472-60493
START_RANGE client302l socket2868 210 60493-60514
START_RANGE client302a socket2869 210 60514-60535
START_RANGE client302b socket2870 220 60535-60557
START_RANGE client302c socket2871 210 60557-60578
START_RANGE client302d socket2872 210 60578-60599
START_RANGE client302e socket2873 210 60599-60620
START_RANGE client302f socket2874 210 60620-60641
START_RANGE client302g socket2875 210 60641-60662
START_RANGE client302h socket2876 210 60662-60683
START_RANGE client302i socket2877 210 60683-60704
START_RANGE client302j socket2878 210 60704-60725
START_RANGE client302k socket2879 210 60725-60746
START_RANGE client302l socket2880 220 60746-60768

#endif
/*----------------------------------------------------------------------*/
#define TES_FLAG_TRACE 0x00000010
#define TES_FLAG_KEYSTROKE_TIME 0x00000200
#define TES_FLAG_LOCAL_LOG 0x00000400
#define TES_FLAG_LOCAL_TRACE 0x00000800
#define TES_FLAG_LOCAL_IPRINT 0x00004000
#if 0
/* SETFLAG ALL TES_FLAG_TRACE */
SETFLAG ALL TES_FLAG_LOCAL_TRACE
SETFLAG ALL TES_FLAG_LOCAL_IPRINT
#endif
#if 0
SETFLAG client31 telnet 1 TES_FLAG_KEYSTROKE_TIME
#endif
```

D.2 user_master.C

```
/*****************************************************************************/
/* user_master.C                                Audit: 05/30/96   */
/*****************************************************************************/

static char *rcsid="$Id: user_master.C,v 1.1 1999/02/22 06:31:05 channui Exp $";

#include <iostream.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#define _H_CUR01
#include <cur00.h>
#undef _H_CUR01
extern "C" {
#include "data/cur01.h"
int wrefresh (WINDOW *);
int wclrtoeol(WINDOW *);
int setupterm(char*,FILE*,int*);
int nodelay(int);
int keypad(int);
int wgetch(WINDOW *);
}
#include "data/rte.h"
#include "data/Stats.h"
#include "data/misc.h"
#include "user_tpcc.h"

struct header_s {
   int slave;
   int num;
   int type;
   int num_timestamps;
   int user_data_length;
   int data_type;
};


char *get_variable(char *name);
int get_variable(char *name, int *number);
int send_global_data(void);
int make_ratios (double *buffer);
extern int ramp_up_complete;
extern int interval_start_time, interval_stop_time;
extern "C" int strcasecmp(char *s1, char *s2);
extern "C" int strncasecmp(char *s1, char *s2, int n);

struct UserSpawnData {
   int Warehouse;
   int District;
};

/* user_master.C */
int user_statistics_print(void);
// int user_spawn(int *length, char *buffer);
int user_spawn(int min, int max, int number, int *length, char *buffer);
int user_finished(intlength, char *buffer);

extern SlaveStatus slave_status[MAX_SLAVES];

extern Stats status[MAX_TRAN_TYPE][MAX_TIMES];
extern WINDOW *statistics_win;
extern UserGlobal *shmglobal;

/* Transaction mix parameters */
double ratio_desired[6], ratio_min[6], ratio_max[6], ratio_range[6];
char *ratio_names[] = { "RTE", "NEWORDER", "PAYMENT", "ORDSTAT", "DELIVERY",
                        "STOCKLEV", NULL };
char *Status_Names[] = {"Menu", "Keying", "Response", "Think"};


char *transaction_names[] = { "RTE", "New Order", "Payment", "Order Stat",
                      "Delivery", "Stock Level", NULL };

static int current_status = 2, status_needs_refresh = 1;

int user_statistics_print(void) {
   int i;
   static int count = 0;
   double ratios[6];
   if (status_needs_refresh) {
                count = 0;
                status_needs_refresh = 0;
                wmove   (statistics_win, 0, 0);
                wprintw (statistics_win, "%11s %8s %8s %8s %8s %8s %6s %6s %6s",
                                Status_Names[current_status], "90%", "Avg", "Min", "Max",
                                "Samples", "Ratio", "Mix", "Think");
   }
   make_ratios(ratios);


   for (i = 1; i <= 5; i++) {
                /* The reason we do this is because calculating the percentiles
                   is expensive */
                if (count % 10 == 0) {
                   wmove   (statistics_win, i, 0);
                   wprintw (statistics_win, "%11s %8.2f",
                                transaction_names[i], status[i][current_status].ninety()/1000.0);
                   count = 0;
                }
                wmove   (statistics_win, i, 21);
                wprintw (statistics_win, "%8.2f %8.2f %8.2f %8d %6.2f %6.2f %6.2f",
                                status[i][current_status].average()/1000.0,
                                status[i][current_status].min()/1000.0,
                                status[i][current_status].max()/1000.0,
                                status[i][current_status].samples(),
                                ratios[i], shmglobal->chances[i],
                                status[i][3].average()/1000.0);
   }
   wmove   (statistics_win, 7, 0);

   extern int runtime_counts[MAX_TRAN_TYPE];
   extern int begin_time, ramp_up, run_time;
   int start = interval_start_time;
   int stop  = interval_stop_time;
   double interval = ((double)(stop-start) / (1000*60));
   double samples = status[1][2].samples();
   if (interval <= 0 || samples <= 0) {
```

```
                wprintw (statistics_win, "TPM-C: %7s / ", "-------");
  } else {
                wprintw (statistics_win, "TPM-C: %7.2f / ", samples/interval);
  }
  samples = runtime_counts[1];
  if (samples > 0) {
                start = begin_time+((ramp_up>=0)?ramp_up:0);
                if (run_time > 0 && stop > begin_time + ramp_up + run_time) {
                   stop = begin_time + ramp_up + run_time;
                }
                interval = (double)(stop - start)/(1000.0*60.0);
                wprintw (statistics_win, "%7.2f", samples/interval);
  } else {
                wprintw (statistics_win, "-------");
  }

  count++;
  return RTE_OK;
}

extern int login_begin;
int login_max_load;

#ifdef WHSEARRAYDBG
int outofboundwarn;
#endif
extern int min_warehouse;
extern int max_warehouse;

const int MAX_WAREHOUSES=100000;
/* All of this 10 stuff is district size.  Should be a constant.
  Maybe fix that later */
int num_warehouses = -1;
int warehouses[MAX_WAREHOUSES*10];
int user_spawn(int min, int max, int number, int *length, char *buffer) {
//int user_spawn(int number, int *length, char *buffer) {
  int i, min_index;
  int adj_wh = num_warehouses;                          // adjusted warehouse number
  UserSpawnData *ptr = (UserSpawnData *)buffer;
  *length = sizeof(*ptr);

//   min_index = 0;
//   for (i = 1; i < (num_warehouses)*10 && i < MAX_WAREHOUSES*10; i++) {
//
//  if both min and max are zero, running START, otherwise running
//                START_RANGE.  Must also determine what the ending warehouse number
//                will be for said range
//

                if (min ==0 && max == 0) {
                                min++;
                                min_index =  0 ;
                } else {
                                adj_wh = max;   // inclusive range of wh-s
                                min = min * 10;
                                min_index = min;
                }
  for (i = min ; i < (adj_wh)*10 && i < ((MAX_WAREHOUSES+min_warehouse)*10); i++) {
                if (warehouses[i - (min_warehouse*10)] < warehouses[min_index - (min_warehouse*10)]) {
                   min_index = i;
                }
  }

  ptr->Warehouse = min_index / 10 + 1;
  ptr->District  = min_index % 10 + 1;
#ifdef WHSEARRAYDBG
  if ((min_index - (min_warehouse*10) < 0) || (min_index - (min_warehouse*10) >=
(MAX_WAREHOUSES*10))) {
    if (outofboundwarn) {
                iprint (IPRINT_INFO, "(spawn) Out of range warehouse number %d, (%d-%d (start) = %d
(rel. num)\n",
                                min_index, min_index, min_warehouse, min_index - (min_warehouse*10));
                outofboundwarn=0;
    }
  }
#endif

  warehouses[min_index - (min_warehouse*10)]++;
/*    iprint (IPRINT_INFO, "Driver for Warehouse %d, District %d started.  warehouses[%d]++ = %d\n",
          ptr->Warehouse, ptr->District, min_index, warehouses[min_index - (min_warehouse*10)]); */
  return RTE_OK;
}

int user_finished(int length, char *buffer) {
  UserSpawnData *ptr = (UserSpawnData *)buffer;
  int temp = (ptr->Warehouse-1)*10+ptr->District-1;

#ifdef WHSEARRAYDBG
  if ((temp - min_warehouse*10 < 0) || (temp - min_warehouse*10 >= MAX_WAREHOUSES*10)) {
    if (outofboundwarn) {
                iprint (IPRINT_INFO, "(finish) Out of range warehouse number %d, (%d-%d (start) = %d
(rel. num)\n",
                                min_index, min_index, min_warehouse, min_index - (min_warehouse*10));
                outofboundwarn=0;
    }
  }
#endif

  warehouses[temp - (min_warehouse*10)]--;
/*    iprint (IPRINT_INFO, "Driver for Warehouse %d, District %d died.  warehouses[%d]-- = %d\n",
          ptr->Warehouse, ptr->District, temp, warehouses[temp - (min_warehouse*10)]); */
  return RTE_OK;
}


double limit(double min, double max, double val) {
  if (val < min)
                return min;
  if (val > max)
                return max;
  return val;
}


int make_ratios (double *buffer) {
  int neword   = status[NEWORDER][0].samples();
  int payment  = status[PAYMENT] [0].samples();
  int ordstat  = status[ORDSTAT] [0].samples();
  int delivery = status[DELIVERY][0].samples();
  int stocklev = status[STOCKLEV][0].samples();
  int total    = neword + payment + ordstat + delivery + stocklev;
  int i;

  if (total == 0) {
                buffer[NEWORDER] = 100.0;
                for (i = 2; i < 6; i++) {
                   buffer[i] = ratio_desired[i];
                   buffer[NEWORDER] -= buffer[i];
                }
                return 0;
  }

  buffer[PAYMENT]  = (double)payment  / (double)total * 100.0;
  buffer[ORDSTAT]  = (double)ordstat  / (double)total * 100.0;
  buffer[DELIVERY] = (double)delivery / (double)total * 100.0;
  buffer[STOCKLEV] = (double)stocklev / (double)total * 100.0;
  buffer[NEWORDER] = 100.0- buffer[PAYMENT] - buffer[ORDSTAT] -
                                                buffer[DELIVERY] - buffer[STOCKLEV];

  return total;
}

int user_global_update(int *length, char *buffer) {
  UserGlobal *shmglobal = (UserGlobal *)buffer;
  static double last[6];
  static last_test_state = 0;
  static int users_last=-1;
  double ratios[6];
  double current[6];
  int i, different = 0;
  int desired = 0;
  int host_busy, all_zero;

  *length = sizeof(*shmglobal);

  make_ratios(ratios);

  /* Calculate ratios we want for next time */
  /* Note: we just keep on with the desired values until ramp-up is complete
        this at least starts us out without any humps or spikes in the
                graph */
  if (ramp_up_complete) {
                current[NEWORDER] = 100.0;
                for (i = 2; i < 6; i++) {
                   if (ratio_desired[i] > ratios[i]) {
                                current[i] = ratio_max[i];
                   } else {
                                current[i] = 2*ratio_desired[i] - ratios[i];
                                if (current[i] < ratio_min[i])
                                   current[i] = ratio_min[i];
                   }
                   current[NEWORDER] -= current[i];
                }
  } else {
                for (i = 1; i < 6; i++) {
                   current[i] = ratio_desired[i];
                }
  }

  /* Add up all the users */
  /* This needs to be changed to be more transparent */
  shmglobal->total_users = 0;
  for (i = 0; i < MAX_SLAVES; i++) {
                shmglobal->total_users += slave_status[i].active;
                desired                += slave_status[i].desired;
  }
  /* Count up number of warehouses we WANT to have */
  if (num_warehouses < 0) {
                num_warehouses = (desired-1)/10+1;
  }
  shmglobal->max_warehouses= num_warehouses;

  host_busy = 0;
  all_zero = 1;
  for (i = 1; i <= 5; i++) {
                if (status[i][current_status].average() != 0) {
                   all_zero = 0;
                }
```

```
                if ( status[i][current_status].average()/1000.0 > login_max_load) {
                    host_busy = 1;
                }
    }
    if (shmglobal->host_busy&& all_zero) {
                host_busy = 1;
    }

    if (host_busy != shmglobal->host_busy){
                shmglobal->host_busy= host_busy;
                different = 1;
    }

    for (i = 2; i < 6; i++) {
                if (current[i] != last[i])
                    different = 1;
    }

    if (last_test_state != shmglobal->test_state) {
                different = 1;
                last_test_state = shmglobal->test_state;
    }

    // Don't send if it's the same as last time
    if ( !different && shmglobal->total_users== users_last ) {
                return RTE_ERROR;
    }

    users_last = shmglobal->total_users;
    for (i = 1; i < 6; i++) {
                shmglobal->chances[i]= last[i] = current[i];
    }

    return RTE_OK;
}

int user_isbusy() {
    return shmglobal->host_busy;
}


int parse_array(char *string, int max, int *buffer) {
    int i, rc;
    char *ptr;
    char *temp = strdup(string);
    ptr = strtok(temp, ",");
    for (i = 0; ptr && i < max; i++) {
                rc = sscanf(ptr, "%d", &buffer[i]);
                if (rc < 1) {
                    free(temp);
                    return i;
                }
                ptr = strtok(NULL, ",");
    }
    free(temp);
    return i;
}

int parse_array(char *string, int max, double *buffer) {
    int i, rc;
    char *ptr;
    char *temp = strdup(string);
    ptr = strtok(temp, ",");
    for (i = 0; ptr && i < max; i++) {
                rc = sscanf(ptr, "%lf", &buffer[i]);
                if (rc < 1) {
                    free(temp);
                    return i;
                }
                ptr = strtok(NULL, ",");
    }
    free(temp);
    return i;
}


int user_init() {
    double  dbuffer[32];
    int   rc, i;
    char *ptr;

    if (get_variable("KEYSTROKE_SLEEP", &shmglobal->keystroke_sleep)!= RTE_OK) {
                shmglobal->keystroke_sleep= 0;
    }
    if (get_variable("LOGIN_TIMEOUT", &shmglobal->login_timeout)!= RTE_OK) {
                shmglobal->login_timeout= 120; /* 2 minutes */
    }
    if (get_variable("KEYSTROKE_PACKET_SIZE",&shmglobal->keystroke_packet_size)!= RTE_OK) {
                shmglobal->keystroke_packet_size= 0;
    }
    shmglobal->login_timeout*= 1000;
    if (get_variable("LOGIN_MAX_LOAD",&login_max_load)!= RTE_OK) {
                login_max_load= 2;
    }
    if (get_variable("WAREHOUSES", &num_warehouses) != RTE_OK) {
                num_warehouses = -1;
    }
    if (get_variable("LASTC", &shmglobal->lastc)!= RTE_OK) {
                shmglobal->lastc= 193; /* 2 minutes */
    }
    iprint(IPRINT_INFO, "Login Timeout = %s\n",
                    mstoa_withfrac(shmglobal->login_timeout,0));
    iprint(IPRINT_INFO, "Keystroke Sleep = %s\n",
                    mstoa_withfrac(shmglobal->keystroke_sleep*1000,0));
    iprint(IPRINT_INFO, "Keystroke Packet Size= %d\n", shmglobal->keystroke_packet_size);
    if (num_warehouses >= 0) {
                iprint(IPRINT_INFO, "Fixed Warehouses to = %d\n", num_warehouses);
    }

    if (!(ptr = get_variable("NEWORDER"))) {
                iprint_error ("Error. NEWORDER variable not found\n");
                exit (1);
    }
    if (parse_array(ptr, 3, dbuffer)!=3) {
                iprint_error ("Error. NEWORDER should be think, emulex_menu, emulex_response");
                exit (1);
    }
    shmglobal->think         [NEWORDER] = dbuffer[0];
    shmglobal->emulex_menu   [NEWORDER] = dbuffer[1];
    shmglobal->emulex_response[NEWORDER]= dbuffer[2];
    shmglobal->test_state = 0;

    for (i = 2; i < 6; i++) {
                if (!(ptr = get_variable(ratio_names[i]))||
                    (parse_array(ptr, 6, dbuffer)!=6)) {
                    iprint(__FILE__, __LINE__, IPRINT_ERROR,
                        "Error. %s should be think, emulex_menu, emulex_response, desired, min, max",
                                ratio_names[i]);
                    exit (1);
                }
                shmglobal->think[i]         = dbuffer[0];
                shmglobal->emulex_menu[i]   = dbuffer[1];
                shmglobal->emulex_response[i]= dbuffer[2];
                ratio_desired[i]            = dbuffer[3];
                ratio_min[i]                = dbuffer[4];
                ratio_max[i]                = dbuffer[5];
                ratio_range[i]              = ratio_max[i]-ratio_min[i];
    }

    for (i=0; i < (MAX_WAREHOUSES*10); i++) {
      warehouses[i] = 0;
    }

#ifdef WHSEARRAYDBG
    outofboundwarn=1;
#endif

    return RTE_OK;
}

int user_extra_data(header_s *header) {
    int i;
    int num_timestamps;

    if (header->data_type != RTE_ITEM_KEYSTROKE_TIMES)
                return RTE_OK;
    int *times = (int *)((char *)header+sizeof(struct header_s));
    num_timestamps= header->user_data_length / 4 - 1;

    iprint (IPRINT_TRACE, "Keystroke times = ");
    for (i = 0 ; i < num_timestamps; i++) {
                iprint (IPRINT_TRACE, "%d ", times[i]);
    }
    iprint (IPRINT_TRACE, "\n", times[i]);

    return RTE_OK;
}

int user_process_command(char *command) {
    char buffer[256], *ptr;
    int i, found, len;
    strncpy (buffer, command, 256);
    ptr = strtok (buffer, " \t");
    found = 0;
    printf ("user_process_command('%s')\n",ptr);
    if (!strcasecmp (ptr, "pause")) {
                shmglobal->test_state = 1;
    } else if (!strcasecmp (ptr, "warmup")) {
                shmglobal->test_state = 2;
    } else if (!strcasecmp (ptr, "notest")) {
                shmglobal->test_state = 0;
    } else if (!strcasecmp (ptr, "login_max_load?")){
                iprint (IPRINT_WARNING, "Current LOGIN_MAX_LOAD = %d\n", login_max_load);
    } else if (!strncasecmp (command, "login_max_load=",15)) {
                login_max_load=atoi(command+15);
                iprint (IPRINT_WARNING, "Set LOGIN_MAX_LOAD = %d\n", login_max_load);
    } else if (!strcasecmp (ptr, "display")) {
        while (ptr && (ptr = strtok(NULL, " \t"))) {
            if (*ptr == '\0')
                continue;
            for (i = 0; i < 5; i++) {
                len = min(strlen(Status_Names[i]), strlen(ptr));
                if (!strncasecmp (ptr, Status_Names[i], len)) {
                    status_needs_refresh = found = 1;
                    current_status = i;
                    return  RTE_OK;
                }
            }
```

```
        iprint (IPRINT_WARNING, "Unknown type to display: %s\n", ptr);
      }
   } else {
                iprint (IPRINT_WARNING, "Unknown Command: '%s'\n", command);
                return RTE_ERROR;
   }
   return RTE_OK;
}

int transaction_process () {
   return RTE_OK;
}

int user_begin() {
   return RTE_OK;
}

/*
void user_make_header(char *buffer) {
   int i;
   struct user_data_header *data = (struct user_data_header *)buffer;
}
*/
```

D.3 user_slave.C

```
/*****************************************************************************/
/* user_slave.C                                  Audit: 05/30/96   */
/*****************************************************************************/

static char *rcsid="$Id: user_slave.C,v 1.1 1999/02/22 06:31:06 channui Exp $";

/*****************************************************************************/
/***                TPCC FILE FOR ALL USERS                        ***/
/*****************************************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/time.h>
#include "rte_slave.h"
#include "user_tpcc.h"

/* This MUST match the corresponding one in client'sinout.h file! */
#define TRIGGER "\021"
//#define NOSLEEP
// Increased EXPECT_TIMEOUT from 600000 - oz 10/20/97
#define EXPECT_TIMEOUT 6000000
#define KEYWAIT_FUDGE  5000

extern SHM_Slave *shm;
extern TableEntrySlave *shmentry;
extern DriverStatus    *status;
extern echo_trace(char *);
extern echo_trace();
extern char *expect_save;

const char *SQL_TPERRNO_MESSAGE = "tperrno";
const char *SQL_RTN_MESSAGE =     "rtn:";
const char *SQL_FATAL_MESSAGE =   "SQL Fatal Error";
const char *ROLLBACK_MESSAGE =    "Item number is not valid";

int            WHSEID;                        /* warehouse number for each users */

/*****************************************************************************/
/* The "uniform()" function has range of the absolute value of the       */
/* difference between the min. and themax values upto 2147483647.      */
/*****************************************************************************/
/*-------------------------------------------------------------*/
/* NURand                          */
/*-------------------------------------------------------------*/
/* A: 255 for C_LAST, 1023 for C_ID, 8191 for OL_I_ID   */
/* x: 0 for C_LAST, 1 for C_ID and OL_I_ID              */
/* y: 999 for C_LAST, 3000 for C_ID, 100000 for OL_I_ID */
/*-------------------------------------------------------------*/
long
NURand(int A, int x, int y, long cval)
{
  return ((((long) uniform((long)0, (long) A) | (long) uniform((long)x, (long) y)) + cval) % (y - x + 1)) + x;
}

/*-------------------------------------------------------------*/
/* getname                         */
/*-------------------------------------------------------------*/
/* generates a random number from 0 to 999 inclusive   */
/* a random name is generated by associating a random  */
/* string with each digit of the generated number      */
/* three strings are concatenated to generate lastname  */
/*-------------------------------------------------------------*/
char          *
getname()
{
  char          *last_name_parts[] =
  {
    "BAR",
    "OUGHT",
    "ABLE",
    "PRI",
    "PRES",
    "ESE",
    "ANTI",
    "CALLY",
    "ATION",
    "EING"
   };
   static char     lastname[128];
   int             random_num;

#if 1
   random_num = NURand(255, 0, 999, shmglobal->lastc);
#else
   random_num = NURand(255, 0, 999, LASTC);
#endif
   strcpy(lastname, last_name_parts[random_num / 100]);
   random_num %= 100;
   strcat(lastname, last_name_parts[random_num / 10]);
   random_num %= 10;
   strcat(lastname, last_name_parts[random_num]);
   return (lastname);
}

typedef struct gen_tran_s {
   int invalid;
   void *data;
   long len;
   long keywait;
   long type;
   char *menu;
   char *request;
} gen_tran_t;

int generic_transaction( gen_tran_t *data ) {
   char buffer[2048];
   static int consecutive_errs = 0;
   int rc;
   set_typing_delay(0);
   iprint(IPRINT_TRACE, "> generic_transaction sleep (%d) type(%d) *data (%d)\n",
data->type,data->menu,data);
#ifndef NOSLEEP
   if (shmglobal->test_state == 0)
                transaction_sleep_do();
#endif

#ifdef EXPECT_TIMEOUT
   int timeout = EXPECT_TIMEOUT;
#else
   int timeout = 0;
#endif

   // Start the transaction (MENU)
   iprint(IPRINT_TRACE, "> generic_transaction start (%d)\n", data->type);
   transaction_start(data->type, data->len, data->data);

   iprint(IPRINT_TRACE, "> transmit data->menu: (%s)\n",data->menu);
   transmit(data->menu);
   echo_trace ("Waiting for Menu");
   if (expect(TRIGGER, timeout) == ERROR) {
                iprint (IPRINT_ERROR, "Slave %d: Failed to receive %s screen\n",
                      shmentry->num, data->menu);
                return (ERROR);
   }
#ifndef NOSLEEP
   usleep(shmglobal->emulex_menu[data->type]*1000000.0+0.9);
#endif

   // Send our request (KEYING)
   transaction_mark(WHERE_NOW);
   echo_trace ("Keying");

#ifndef NOSLEEP
   usleep(data->keywait*1000000+KEYWAIT_FUDGE);// Keying delay
#endif
   // Wait for response (RESPONSE)
   transaction_mark(WHERE_NOW);

   iprint(IPRINT_TRACE, "> transmit data->request\n");
   transmit(data->request);

   echo_trace ("Wait for Response");
   if (expect(TRIGGER, timeout) == ERROR) {
                iprint (IPRINT_ERROR, "Slave %d: Failed to receive %s response\n",
                              shmentry->num, data->menu);
                return (ERROR);
   }
#ifndef NOSLEEP
   usleep(shmglobal->emulex_response[data->type]*1000000.0+0.9);
#endif

   // Look for errors and set our think time (THINK)
   transaction_mark(WHERE_NOW);
   if (expect_after_match ("ERROR: ")) {
                data->invalid = 1;
                iprint (IPRINT_ERROR, "Slave %d: %s found '%s'\n",
```

```
                        shmentry->num, data->menu, "ERROR:");
                // Very dangerous, keep going rather than exiting...
                // return RTE_ERROR;
                // Check for consecutive errors and if there are more than
                // 4 of them exit - allow for transient errors to make
                // tuning and testing easier   -oz
                // In either case the transaction is marked as invalid and
                // will be reported as an error by the analyze program.
                if (consecutive_errs++ > 4)
                    return RTE_ERROR;
        } else {
                consecutive_errs = 0;
        }
        echo_trace ("Thinking");
        transaction_sleep_set(neg_exp_4(shmglobal->think[data->type])*1000.0);
        iprint(IPRINT_TRACE, "< generic_transaction finish\n");
        return (RTE_OK);
}

/*****************************************************************************/
/***                 Delivery Transaction                                  ***/
/*****************************************************************************/
int
Delivery()
{
        static struct delivery_struct delivery, delivery_new;
        int             rc;
        char            *ptr;
        char                buffer[256];
        gen_tran_t      tran;

        tran.invalid = 0;
        tran.data    = &delivery;
        tran.len     = sizeof(delivery);
        tran.keywait = 2;
        tran.type    = DELIVERY;
        tran.menu    = "4";
        tran.request = buffer;

        // Set up all data for new transactions
        delivery_new.carrier = uniform(1, 10);       // carrier # 1 to 10

        // Now create the actual request
        ptr = buffer;
        ptr += sprintf(ptr, "%d\n", delivery_new.carrier);

        // Go do the transaction
        rc = generic_transaction(&tran);
        delivery        = delivery_new;
        delivery.invalid= tran.invalid;

        return (rc);
}

/*****************************************************************************/
/***                 New Order Transaction                                 ***/
/*****************************************************************************/
int NewOrder() {
        static struct neword_struct neword, neword_new;
        int             i, rc, whses, low_whse=1;
        char            buffer[2048];
        char            *ptr;
        const char      *ptr2;
        gen_tran_t      tran;

        tran.invalid = 0;
        tran.data    = &neword;
        tran.len     = sizeof(neword);
        tran.keywait = 18;
        tran.type    = NEWORDER;
        tran.menu    = "1";
        tran.request = buffer;

        neword_new.rollback=0;

        /*** SECTION TO DETERMINE ROLLBACK TRANSACTION FOR 1% OF NEW ORDERS ***/
        neword_new.did = uniform(1, 10);                        // district number
        neword_new.cid = NURand(1023, 1, 3000, CUSTC);    // customer # 1 to 3000
        neword_new.nloop = uniform(5, 15);          // number of items to order (5-15)
        neword_new.olremote=0;   // find total number of remote order-lines

        whses = shmglobal->max_warehouses;

        for (i = 0; i < neword_new.nloop; i++) {
                // Warehouse Number
                neword_new.item[i].olswid = WHSEID;
                if (whses > 1 && (uniform(0.0, 100.0) < 1.0)) {
                    /* for 1% of items (if * uniform()==0) */
                    /* Generate a uniform whse number that's different from WHSEID */
                    neword_new.item[i].olswid =
                                (long) uniform((long) low_whse, (long)whses-1);
                    if (neword_new.item[i].olswid >= WHSEID)
                                neword_new.item[i].olswid++;
                    neword_new.olremote++;  // find total number of remote order-lines
                }
                // Item number 1-100000
                neword_new.item[i].oliid = NURand(8191, 1, 100000, ITEMC);
                // Quantity 1-10
                neword_new.item[i].olquantity = uniform(1, 10);
        }                                                       /* end of for n_loop */
        // We occasionally force a transaction to have invalid data to force a
        // rollback
        if (uniform(1, 5000) <= 50)
                neword_new.item[neword_new.nloop-1].oliid = 999999;

        neword_new.oremote = (neword_new.olremote > 0);

        // Now create the actual request
        ptr = buffer;
        ptr += sprintf(ptr, "%d\t%d", neword_new.did, neword_new.cid);
        for (i = 0; i < neword_new.nloop; i++) {
                ptr += sprintf(ptr, "\t%d\t%d\t%d",
                                        neword_new.item[i].olswid,
                                        neword_new.item[i].oliid,
                                        neword_new.item[i].olquantity);
        }
        ptr += sprintf(ptr, "\n");

        // Go do the transaction
        rc = generic_transaction(&tran);
        neword      = neword_new;
        neword.invalid = tran.invalid;

        // Check for a rollback
        if (expect_after_match (ROLLBACK_MESSAGE)) {
                neword.rollback=1;
                echo_trace ("Found rollback!\n");
        }

        // Grab the orderID from the
        if (!(ptr2 = expect_after_match("\033[6;15H"))) {
                echo_trace ("Didn't find order-id for neworder");
                iprint (IPRINT_ERROR, "Neworder didn't have Order-ID");
                neword.oid = -1;
        } else {
                neword.oid = atoi(ptr2+8);
        }

        // This is really not useful since we aren't going to be sending individual
        // keystrokes anymore
        if (shmentry->flags & TES_FLAG_KEYSTROKE_TIME) {
                log_data(RTE_ITEM_KEYSTROKE_TIMES,
keystroke_length*sizeof(int),keystroke_times);
        }

        return (rc);
}


/*****************************************************************************/
/***                 Order Status Transaction                              ***/
/*****************************************************************************/
int OrderStatus() {
        static struct ordstat_struct ordstat, ordstat_new;
        char            buffer[2048];
        int             rc;
        char            *ptr;
        gen_tran_t      tran;

        tran.invalid = 0;
        tran.data    = &ordstat;
        tran.len     = sizeof(ordstat);
        tran.keywait = 2;
        tran.type    = ORDSTAT;
        tran.menu    = "3";
        tran.request = buffer;

        // Set up all data for new transactions
        ordstat_new.did = uniform(1, 10);               /* district number 1 to 10 */
        if (uniform(1, 100) <= 60) {/* for 60% of transactions */
                char *tmp = getname();
                strcpy(ordstat_new.clast, tmp);                 /* by customer last name */
                if (ordstat_new.clast[0] < 'A' || ordstat_new.clast[0] > 'Z') {
                    iprint (IPRINT_ERROR,
                        "ASSERTION: OrderStatus getname() returns invalid name!'%s'\n",
                                ordstat_new.clast);
                    return RTE_ERROR;
                }
                ordstat_new.byname = 1;
                ordstat_new.cid = 0;
        } else {
                ordstat_new.cid = NURand(1023, 1, 3000, CUSTC);         /* cust. # 1 to 3000 */
                ordstat_new.byname = 0;
                ordstat_new.clast[0] = (char) NULL;
        }

        // Now create the actual request
        ptr = buffer;
        ptr += sprintf(ptr, "%d\t", ordstat_new.did);
        if (ordstat_new.byname) {
                ptr += sprintf(ptr, "\t%s\n", ordstat_new.clast);
        } else {
                ptr += sprintf(ptr, "%d\n", ordstat_new.cid);
        }

        // Go do the transaction
        rc = generic_transaction(&tran);
        ordstat      = ordstat_new;
```

```
  ordstat.invalid = tran.invalid;

  return (rc);
}

/*****************************************************************************/
/***                  Payment Transaction                                  ***/
/*****************************************************************************/
int
Payment()
{
  static struct payment_struct payment, payment_new;
  int           dollars, cents,  rc, whses, low_whse = 1;
  char          buffer[2048];
  char          *ptr;
  gen_tran_t    tran;

  tran.invalid = 0;
  tran.data    = &payment;
  tran.len     = sizeof(payment);
  tran.keywait = 3;
  tran.type    = PAYMENT;
  tran.menu    = "2";
  tran.request = buffer;

  payment_new.did = uniform(1, 10);           /* district number 1 to 10 */
  if (uniform(1, 100) <= 60) {/* for 60% of transactions */
                strncpy(payment_new.clast, getname(), 17);    // by customer last name
                if (payment_new.clast[0] < 'A' || payment_new.clast[0] > 'Z') {
                   iprint (IPRINT_ERROR,
                      "ASSERTION: payment_new getname() returns invalid name!'%s'\n",
                            payment_new.clast);
                   return RTE_ERROR;
                }
                payment_new.byname = 1;
                payment_new.cid = 0;
  } else {
                payment_new.cid = NURand(1023, 1, 3000, CUSTC);       /* cust. # 1 to 3000 */
                payment_new.byname = 0;
                payment_new.clast[0] = (char) NULL;
  }

  whses = shmglobal->max_warehouses;

  if (whses < 2 || uniform(1, 100) <= 85) {/* for 85 % of transactions */
                payment_new.cwid = WHSEID;
                payment_new.cdid = payment_new.did;
                payment_new.remote = 0;
  } else {                                  /* for 15 % of transactions */
                payment_new.cwid = (long) uniform((long)low_whse,(long) whses-1);
                if (payment_new.cwid >= WHSEID)
                   payment_new.cwid++;

                payment_new.remote = 1;
                payment_new.cdid = uniform(1, 10);          /* district 1 to 10 */
  }

  dollars = uniform(1, 5000);  /* dollar amt = 1 to 5000 */
  if (dollars == 5000)
                cents = 0;
  else
                cents = uniform(0, 99);

  payment_new.amount = ((double) dollars) + ((double) cents) / 100.0;

  // Now create the actual request
  ptr = buffer;
  ptr += sprintf(ptr, "%d\t", payment_new.did);
  if (payment_new.byname){
                ptr += sprintf(ptr, "\t%s\t", payment_new.clast);
  } else {
                ptr += sprintf(ptr, "%d\t\t", payment_new.cid);
  }
  ptr += sprintf(ptr, "%d\t%d\t", payment_new.cwid, payment_new.cdid);
  ptr += sprintf(ptr, "%d.%02.2d\n", dollars, cents);

  // Go do the transaction

  rc = generic_transaction(&tran);
  payment        = payment_new;
  payment.invalid= tran.invalid;

  return (rc);
}

/*****************************************************************************/
/***                  Stock Level Transaction                              ***/
/*****************************************************************************/
int
StockLevel()
{
  static struct stocklev_struct stocklevel, stocklevel_new;
  char          buffer[2048];
  int           rc;
  char          *ptr;
  gen_tran_t    tran;

  tran.invalid = 0;
  tran.data    = &stocklevel;
  tran.len     = sizeof(stocklevel);
  tran.keywait = 2;
  tran.type    = STOCKLEV;
  tran.menu    = "5";
  tran.request = buffer;

  stocklevel_new.invalid= 0;
  stocklevel_new.threshold = uniform(10, 20);                 /* uniform no. between 10 and
                                                                                  * 20 */
  // Now create the actual request
  ptr = buffer;
  ptr += sprintf(ptr, "%d\n", stocklevel_new.threshold);

  // Go do the transaction
  rc = generic_transaction(&tran);
  stocklevel        = stocklevel_new;
  stocklevel.invalid= tran.invalid;

  return (rc);
}


/*****************************************************************************/
/***  MAIN()                                                               ***/
/*****************************************************************************/
int
user_transaction()
{
  char          logout[32];
  double        ntask;
  int           resp;
  static int task = 0;

  if (shmentry->flags& TES_FLAG_KEYSTROKE_TIME) {
                int rc;
                /* Wait for specified period of time*/
                sleep (shmglobal->keystroke_sleep);
                /* Quit after one transaction */
                shm->lock(shmentry->pid);
                shmentry->flags|= TES_FLAG_DIE;
                shm->unlock(shmentry->pid);
                rc = NewOrder();
                iprint (IPRINT_INFO, "Slave %d: Keystroke timing setting dieflag\n", shmentry->num);
                return rc;
  }

#if 1
  switch (shmglobal->test_state) {
  case 0: // Normal
                break;
  case 1: // pause
                sleep (1);
                return RTE_OK;
  case 2: // warmup
                switch(task++) {
                case 0: return Delivery();
                case 1: return OrderStatus();
                case 2: return Payment();
                case 3: return StockLevel();
                case 4: task = 0; return NewOrder();
                }
  }
  /*****************************************************************************/
  /***  CHOOSE ONE OF THE TRANSACTIONS                                       ***/
  /*****************************************************************************/
  ntask = (double) uniform(0.0, 100.0);
  if (ntask <= shmglobal->chances[DELIVERY]){
                return Delivery();
  }
  ntask -= shmglobal->chances[DELIVERY];
  if (ntask <= shmglobal->chances[ORDSTAT]){
                return OrderStatus();
  }
  ntask -= shmglobal->chances[ORDSTAT];
  if (ntask <= shmglobal->chances[PAYMENT]){
                return Payment();
  }
  ntask -= shmglobal->chances[PAYMENT];
  if (ntask <= shmglobal->chances[STOCKLEV]){
                return StockLevel();
  }
  return NewOrder();
#else
  // this code should be shared between all of the users on a slave
  // int the best case it should be shared between all of the slaves,
  // but that would be too costly.
  // for now it is done on a per user basis.  If this thing is ever
  // modified to be threaded then it will probably go to the per-process
  // basis.  Although with shared memory, it would be possible to go to
  // per-slave.  Actually, before this code is put into use it must be
  // fixed up to share across processes.  Right now it will take, on average,
  // 22 minutes for one user to just key in the 100 entries.

  // use a card deck with no replacement to fulfill the requirements
  {
                int deck[100], count=-1, i, size=1, tmp;
                // lock deck
                if (count < 0) {
                   // deck is empty fill it up
```

```
                count = 0;
                for (i = 0; i < 43 * size; i++) {
                        deck[count++] = Payment;
                }
                for (i = 0; i < 4 * size; i++) {
                        deck[count++] = StockLevel;
                }
                for (i = 0; i < 4 * size; i++) {
                        deck[count++] = OrderStatus;
                }
                for (i = 0; i < 4 * size; i++) {
                        deck[count++] = Delivery;
                }
                for (; count < 100 * size; i++) {
                        deck[count++] = NewOrder;
                }
                // randomize the deck
                for (i = 0; i < 100 * size; i++) {
                        int tmp;
                        int pick = uniform(i+1, 100);
                        tmp = deck[i];
                        deck[i] = deck[pick];
                        deck[pick] = tmp;
                }
        }
        tmp = deck[count--];
        // unlock deck
        switch(tmp) {
        case Delivery:    return Delivery();
        case OrderStatus: return OrderStatus();
        case Payment:     return Payment();
        case StockLevel:  return StockLevel();
        case NewOrder:    return NewOrder();
        }
  }
#endif

#if 0
  if (resp != RTE_OK) {                 /* logoff if response is not correct */
                strcpy(logout, "9\n");          /* menu option 9 */
                transmit(logout);
                resp = expect("tpcc_cstux_inf:");
                return (ERROR);
  } else
                return (RTE_OK);
#endif
}                                                       /* end of main */

int user_parameter_change(void) {
  #if 0
  int i;
  iprint(IPRINT_TRACE, "Slave %d: total_users = %d\n", shmentry->num);
  iprint(IPRINT_TRACE, "Slave %d: chances     = ", shmentry->num);
  for (i = 0; i < MAX_TRAN_TYPE; i++)
                iprint(IPRINT_TRACE, "%6.2f ", shmglobal->chances[i]);
  iprint(IPRINT_TRACE, "\nSlave %d: think      = ", shmentry->num);
  for (i = 0; i < MAX_TRAN_TYPE; i++)
                iprint(IPRINT_TRACE, "%6.2f ", shmglobal->think[i]);
  iprint(IPRINT_TRACE, "\n");
  #endif
  return RTE_OK;
}

int user_login(char *user, char *password, void *data) {
  UserLocal *localdata = (UserLocal *)data;
  int rc;
  int timeout_value = shmglobal->login_timeout;
  char buffer[32];
  set_typing_delay(0);

  rc = expect   (TRIGGER, timeout_value);
  if (rc == RTE_ERROR) {
                iprint (IPRINT_ERROR, "Slave %d: didn't find Warehouseprompt\n", shmentry->num);
  }
  sprintf(buffer, "%d\t%d\n", localdata->Warehouse, localdata->District);
  transmit(buffer);
  iprint (IPRINT_TRACE, "Slave %d: Warehouse=%d, District=%d, pid=%d\n", shmentry->num,
localdata->Warehouse, localdata->District, getpid());

  rc = expect (TRIGGER, timeout_value);
  if (rc != RTE_OK) {
     iprint (IPRINT_ERROR, "Slave %d: Failed loggingin\n", shmentry->num);
                return RTE_ERROR;
  }
  return RTE_OK;
}

int user_init () {
  extern int expect_save_active;
  WHSEID = shmlocal->Warehouse;

  status->max_transmit =  shmglobal->keystroke_packet_size;
  expect_save_active = 1;
  return RTE_OK;
}

int user_logout () {
  transmit("9");
  iprint (IPRINT_TRACE, "Slave %d: Warehouse=%d, District=%d\n", shmentry->num,
shmlocal->Warehouse, shmlocal->District);
  return RTE_OK;
}

int user_cleanup () {
  transaction_sleep_do();
  transaction_start(0, 0, NULL); // Just something to clear out the buffer...
  return RTE_OK;
}

int user_spawn_ok() {
  int rc, hb;
  hb = ((UserGlobal *)(shm->global_data))->host_busy;
  rc = hb?RTE_ERROR:RTE_OK;
  return rc;
}
```

D.4 user_tpcc.h

```
 /*****************************************************************************/
/* user_tpcc.h                                   Audit: 05/30/96   */
/*****************************************************************************/

/* $Id: user_tpcc.h,v 1.1 1999/02/22 06:31:06 channui Exp $ */

#ifndef USER_TPCC_H
#define USER_TPCC_H

#include "data/rte_common.h"

/*****************************************************************************/
/***   run-time constant for customer last name from 0 to 255,          ***/
/***   run-time constant for customer id from 0 to 1023,                ***/
/***   run-time constant for item id from 0 to 8191.                    ***/
/*****************************************************************************/
/* #define      LASTC       117 */
/* Change for 3.1 */
#define         LASTC       193
#define         CUSTC       319
#define         ITEMC       3849

/*****************************************************************************/
/***   response type
***/
/*****************************************************************************/
/* #define OK  1  */
/* #define ERROR               -1 */

/*****************************************************************************/
/***   transaction type
***/
/*****************************************************************************/
#define         NEWORDER 1
#define         PAYMENT                 2
#define         ORDSTAT                 3
#define         DELIVERY   4
#define         STOCKLEV   5

/*****************************************************************************/
/***   transaction structures
***/
/*****************************************************************************/
struct neword_struct {
  char          invalid;        /* transaction completed sucessfully */
  long          did;
  long          cid;
  long          oid;    /* Order-ID returned from client */
  long          nloop;          /* number of order line, avg = 15 */
  char          oremote;        /* 1 for remote order, 10%  */
  long          olremote;       /* number of remote order line, 1% */
  char          rollback;       /* actually saw rollback text on screen */
  struct items_struct {
                long          olswid;
                long          oliid;
                long          olquantity;
  } item[15];
};

struct payment_struct {
  char          invalid;        /* transaction completed sucessfully */
  long did;
  long cid;
  long cwid;
  long cdid;
  char clast[17];
  double amount;
  char byname;                                  /* 1 for by last name, 0 for by id*/
  char remote;                          /* 1 for remote wardhouse, 0 otherwise */
};

struct ordstat_struct {
```

```
  char          invalid;          /* transaction completed sucessfully*/
  long did;
  long cid;
  char clast[17];
  char byname;                                          /* 1 for by last name, 0 for by id*/
};

struct delivery_struct {
  char          invalid;          /* transaction completed sucessfully*/
  char carrier;
};

struct stocklev_struct {
  char          invalid;          /* transaction completed sucessfully*/
  long threshold;
};

struct generic_struct {
  char          invalid;          /* transaction completed sucessfully*/
};

typedef union transaction_info {
  char   invalid;
  struct generic_struct  generic;
  struct neword_struct   neword;
  struct payment_struct  payment;
  struct ordstat_struct  ordstat;
  struct delivery_struct delivery;
  struct stocklev_struct stocklev;
} transaction_info;

struct UserGlobal {
  int total_users;
  int max_warehouses;
  int keystroke_sleep;
  int login_timeout;
  int keystroke_packet_size;
  int lastc;
  int test_state;
  int host_busy;
  double chances[MAX_TRAN_TYPE];
  double think[MAX_TRAN_TYPE];
  double emulex_response[MAX_TRAN_TYPE];
  double emulex_menu   [MAX_TRAN_TYPE];
};

struct UserLocal {
  int Warehouse;
  int District;
};

/*
struct user_data_header {
};
*/

extern struct UserGlobal *shmglobal;
extern struct UserLocal  *shmlocal;

#endif
```

APPENDIX E: Third Party Quotes

FS108NA 10/100 8PORT DUAL SPEED SWITCH RJ45 W/ UPLINKBUTTON

Part Number : FS108NA
In Stock : NO
Platform : PC Hardware
Media : Peripherals

Price: $84.99

BuyNow

Description:

Eight port Workgroup Fast Ethernet Switch which improves Network traffic by Segmentation. The FS108 Fast Ethernet Switch brings the 100 Mbps switching technology in a compact form factor to the small office marketplace at an aggressive price. This switch segments the network into smaller, connected subnets for improved performance, enabling the most demanding multimedia and imaging applications. Since each port is auto-speed-sensing, individual subnets or directly attached servers can easily be upgraded from 10 to 100 Mbps.

Features:

- Enhance productivity and network reliability by segmenting the workgroup into smaller units. Eight auto-sensing UTP ports One speed-sensing UTP port selectable between Normal or Uplink connection. Stores up to 4,096 MAC addresses per system. Filtering and Forwarding Rates. 14,800 packets per second for 10 Mbps ports 148,000 packets per second for 100 Mbps ports Half/full duplex switch.

APPENDIX F: Pricing Information

The price of the IBM-supplied configuration will be available for at least sixty days. Prices are in US Dollars and are for sale within the United States of America. Prices in other countries may vary. Discounts are determined based on the overall value of the specific components in this single quotation, without requirement for any past, future, or additional purchase. Discounts for similarly sized configurations will be similar to those quoted here, but may vary based on the components in the configuration.

For assistance regarding these prices or their applicability to any customer's requirements, please contact one of the following individuals:

Bill Casey,
EServer pSeries Offering Manager
wrcasey@us.ibm.com
(512) 838-1422

MaryBeth Pierantoni,
Mary.beth.pierantoni@oracle.com
(650) 506-2118

Thank you,

Robert Schramer,
Mgr, Product Pricing & Business Evaluation - eServer pSeries
schramer@us.ibm.com
(512) 838-5297