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

Using

AIX 5L V5.2

and

DB2 Universal Database 8.1

TPC Benchmark™ C

Full Disclosure Report

Submitted For Review

May 9, 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

DB2, DB2 UDB, DB2 Universal Database

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

TPC Benchmark, TPC-C, and tpmC	Trademark of the Transaction Processing Performance Council
TUXEDO 8.0	Trademark of BEA System, Inc

First Edition May 9, 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:

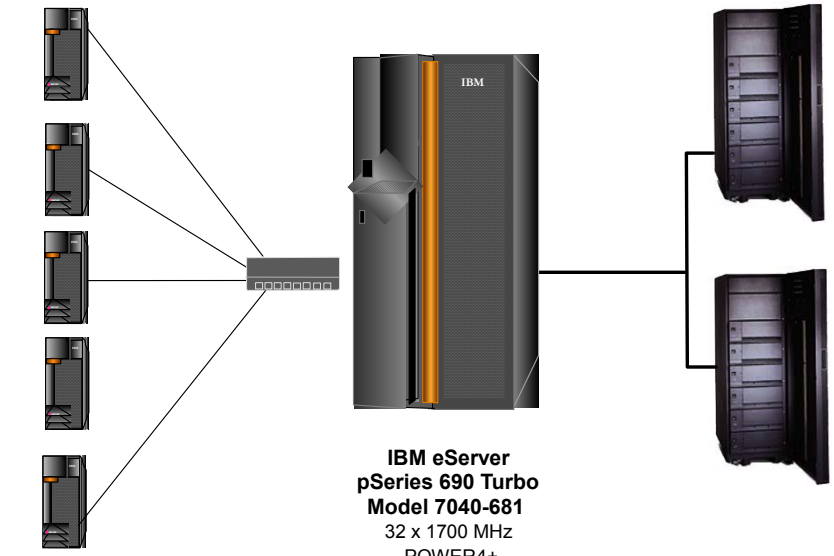
TPC Benchmark Administrator
IBM Commercial Performance
Mail Stop 9571
11501 Burnet Road
Austin, TX 78758
FAX Number (512) 838-1852

© Copyright International Business Machines Corporation 2003. All rights reserved.

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

NOTE: US. Government Users - Documentation related to restricted rights: Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

IBM	IBM eServer pSeries 690 Turbo Model 7040-681		TPC-C Rev. 5.1	
			Report Date: May 9, 2003	
Total System Cost	TPC-C Throughput	Price/Performance	Availability Date	
$7,574,961	680,613.12 tpmC	$11.13/tpmC	November 8, 2003	
Processors	Database Manager	Operating System	Other Software	No. Users
32 for database 108 for clients	DB2 UDB 8.1	AIX 5L V5.2	TUXEDO 8.0	532,000



System Components	Clients		Server	
	Quantity	Description	Quantity	Description
Processor	27	4 x1000 MHz POWER4 W/32MB L3 cache each	1	32 x 1700 MHz POWER4+ w/ 128 MB L3 cache per MCM, 4 MCM in the SUT
Memory	16 GB		512 GB	
Disk Controllers	2	Ultra3 SCSI-2 (Integrated)	1 30	SCSI-2 (Integrated) SSA Adapters
Disk Drives	1	18.2 GB Ultra SCSI each client	2,560 3	18.2 GB SSA disk 18 .2 GB SCSI disk
Total Storage	16.96 GB each client		43,067 GB	
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, 600MHz	4199	1	25,000	4	100,000	0
64GB H-Memory (inward facing)	4488	1	172,000	4	688,000	0
64GB H-Memory (outward facing)	4489	1	172,000	4	688,000	0
1.7GHz 8-way Processor MCM	5246	1	350,000	4	1,400,000	372,000
Std CEC Fan Pwr cbl grp, BPC(F&R) to Std CEC Fans	6161	1	400	1	400	0
Std CEC Service Proc Y-Cbl, BPC(F&R) to Std CEC SF	6162	1	200	1	200	0
Standard CEC DC/DC Converter Assembly	6170	1	4,700	2	9,400	0
Power Cable Group, 1 thru 3 processor modules	6181	1	720	1	720	0
Additional DC/DC Converter Assembly, (DCA)	6189	1	4,200	3	12,600	0
Capacitor book (for 2 MCMs)	6198	1	1,800	2	3,600	0
Service Processor + Dual Rio Loop (GX)	6418	1	7,000	1	7,000	0
Remote I/O Loop adapter	6410	1	8,000	1	8,000	0
Backplane, Central Electronics Complex	6565	1	50,000	1	50,000	0
Attachment Cable, Hardware Management Console	8121	1	75	1	75	0
Media Drwr, 1U, Op Pnl/Dskt/4 SCSI Media Bays	8692	1	1,700	1	1,700	0
I/O Drawer	7040-61D	1	3,980	4	15,920	16,320
SCSI cable - B&C to Media Drawer	2122	1	275	1	275	0
RIO-G Cable, 500MHz, 1.2m + 2x3.5M	3146	1	1700	4	6,800	0
18.2 GB Ultra3 10K RPM 1" U3 Module	3157	1	1,260	3	3,780	0
10/100 Mbps Ethernet adapter II	4962	1	275	10	2,750	0
Cable Grp, 4xUPIC/2xRIO, BPM(IO#1 thru#4) to B&C	6121	1	1600	1	1,600	0
I/O Drwr DC/DC Converter assembly, (DCA)	6172	1	4000	8	32,000	0
Power Cable, B&C to Media Drawer	6179	1	300	1	300	0
Ultra SCSI SE Adapter	6206	1	395	1	395	0
Advanced SerialRAID Adapter	6230	1	3000	30	90,000	0
32 MB Fast_Write Cache option card	6235	1	575	2	1,150	0
B&C Planar, 10 PCI Slots, 2x integrated SCSI	6563	1	8,000	8	64,000	0
Ultra3 SCSI 4-Pack	6564	1	500	16	8,000	0
Rack	7040-61R	1	5,500	1	5,500	3,960
Front Door, 24W, 2M, and back door	6070	1	3750	1	3,750	0
Bulk Power Regulator (BPR)	6186	1	4000	4	16,000	0
Bulk Power Controller (BPC), 4 Fans + 3 DCAs	6187	1	1900	2	3,800	0
Bulk Power Distribution (BPD), 10 DCAs	6188	1	3500	4	14,000	0
Line Cord, 60A, 14', IEC309 Plug, Chargeable	8678	1	1000	2	2,000	0
Regatta-H Bulk Power Module (2x BPA, EPO PNL)	8690	1	5000	1	5,000	0
System Console, serial attach. cabel 6M	7316	1	4,400	1	4,400	0
Display, White P76/P77 Color Monitor, keyboard	3627	1	675	1	675	0
				Subtotal	**3,265,184**	**413,040**
Storage						
SSA Disk Subsystem	7014-T42	1	4,150	16	66,400	14,208
Front Trim, 2 side Trims	6081	1	450	16	7,200	0
Additional Power Distribution Unit	6171	1	1,000	43	43,000	0
				Subtotal	**116,600**	**14,208**
2.6 TB SSA Disk Bundle	7133-1BU	1	587,756	1	587,756	44,160
13.1 TB SSA Disk Bundle	7133-5BU	1	2,351,025	1	2,351,025	220,800
26.2 TB SSA Disk Bundle	7133-9BU	1	3,918,375	1	3,918,375	441,600
				Subtotal	**6,857,156**	**706,560**

Server Software						
AIX 5.2 (media only)	5962-A5L	1	50	1	50	0
AIX SWMA Subsription	5771-SUB	1	275	32	0	8,800
AIX SWMA Support	5771-SPT	1	775	32	0	24,800
C for AIX user Lic+SW maint 12 MO	D5A1DLL	1	515	1	515	0
C for AIX user annual SW maint renewal	E1A1FLL	1	103	2	0	206
DB2 Enterprise Server Edition Proc Lic/1 yr Maint.	D158GLL	1	19,755	32	632,160	0
DB2 Enterprise Server Ed Proc Maint Renew (annual).	E00BILL	1	940	64		60,160
				Subtotal	**632,725**	**93,966**
Client Hardware and Software						
Deskside Server 1:pSeries 630 Model 6E4, CD ROM	7028-6E4	1	808	27	21,816	81,000
18.2 GB 10,000 RPM Ultra3 SCSI Disk Drive, cable	3157	1	1,112	27	30,024	0
P76/p77 Color Monitor, captured cable, keyboard	3627	1	562	1	562	0
4096MB (4x1024MB) SDRAM DIMM Memory	4453	1	8,800	108	950,400	0
4-port 10/100 Mbps Ethernet adapter	4961	1	1,250	135	168,750	0
2-way 1.0 GHz POWER4 processor card	5132	1	13,000	54	702,000	0
Pwr Supply ,645 W AC, Hot-Swap, base and redunadar	6273	1	2,083	27	56,241	0
Ultra3 SCSI 4-Pack Hot Swap Back Plane	6568	1	292	27	7,884	0
AIX SWMA Subsription	5771-SUB	1	175	2	0	350
AIX SWMA Support	5771-SPT	1	475	2	0	950
AIX 5L version 5	5962-A5L	1	50	27	1,350	0
				Subtotal	**1,939,027**	**82,300**
Third Party Hardware/Software						
BEA Tuxedo 8.0		2	10,200	27	275,400	173,502
10/100 16-Port Dual Speed Switch	FS116NA	3	98	5	490	0
				Subtotal	**275,890**	**173,502**
Audited by: Francois Raab, Info Sizing (www.infosizing.com)				**Discounts**	**-6,475,282**	**-519,915**
Pricing Sources:				**Total**	**6,611,300**	**963,661**
1=IBM HW: Bill Casey, eServer pSeries Offering Mgr, wrcasey@us.ibm.com, 512-838-1422						
1= IBM DB2:Paul Rivot, Director Database Servers and				**Three-Year Cost of Ownership**		**US$7,574,961**
Business Intelligence Software, e-mail privot@us.ibm.com, 914-766-1325					**tpmC**	**680,613.12**
2= BEA Systems					**$/tpmC**	**US$11.13**
3=NETGEAR						

See next page for details of the SSA Disk Rack Bundles.

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 se ctions 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

2.6 TB SSA Disk Bundle	7133-1BU			US$587,756
	Pre-configured bundle of parts below			
	Part No.	**Qty**		
SSA Disk Subsystem	7133-D40	9		
50/60 Hz AC, 300 VDC Power Supply	8022	9		
Raven Black Drawer Cover	8031	9		
10K/18.2 GB Advanced Disk Drive Module	8518	144		
SSA Cables	8805	9		

13.1 TB SSA Disk Bundle	7133-5BU			US$2,351,025
	Pre-configured bundle of parts below			
	Part No.	**Qty**		
SSA Disk Subsystem	7133-D40	45		
50/60 Hz AC, 300 VDC Power Supply	8022	45		
Raven Black Drawer Cover	8031	45		
10K/18.2 GB Advanced Disk Drive Module	8518	720		
SSA Cables	8805	45		

26.2 TB SSA Disk Bundle	7133-9BU			US$3,918,375
	Pre-configured bundle of parts below			
	Part No.	**Qty**		
SSA Disk Subsystem	7133-D40	90		
50/60 Hz AC, 300 VDC Power Supply	8022	90		
Raven Black Drawer Cover	8031	90		
10K/18.2 GB Advanced Disk Drive Module	8518	1,440		
SSA Cables	8805	90		

MQTH, computed Maximum Qualified Throughput: 680,613.12 ttpmC

Response Times (in seconds)	90th %	Average	Maximum
New Order	0.12	0.08	17.99
Payment	0.08	0.05	24.02
Order-Status	0.13	0.09	12.45
Delivery (interactive)	0.05	0.03	14.26
Delivery (deferred)	0.28	0.23	11.76
Stock-Level	0.08	0.05	11.92
Menu	0.00	0.00	15.11

Transaction Mix, in percent of total transactions	Percent
New Order	44.91%
Payment	43.02%
Order-Status	4.02%
Delivery	4.01%
Stock-Level	4.01%

Keying/Think Times (in seconds)	Min.	Average	Max.
New Order	18.00/0.01	18.01/12.02	18.12/120.20
Payment	3.00/0.01	3.01/12.02	3.08/120.21
Order-Status	2.00/0.01	2.01/10.01	2.08/100.10
Delivery	2.00/0.01	2.01/5.02	2.08/50.20
Stock-Level	2.00/0.01	2.01/5.02	2.08/50.20

Test Duration	
Ramp-up Time	29 minutes 19 seconds
Measurement interval	4 hours 30 minutes
Transactions during measurement interval (all types)	409,144,518
Ramp-down time	17 minutes

Checkpointing	
Number of checkpoints	N/A
Checkpoint interval	N/A

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, DB2 UDB 8.1 database manager, and Tuxedo 8.0 for AIX transaction manager.

IBM eServer pSeries 690 Turbo Model 7040-681

Company Name	System Name	Data Base Software	Operating System Software
IBM Corporation	IBM eServer pSeries 690 Turbo Model 7040-681	DB2 UDB 8.1	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
$7,574,961	680,613.12	$11.13 per 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 and DB2 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 eServer pSeries 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.**

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.*

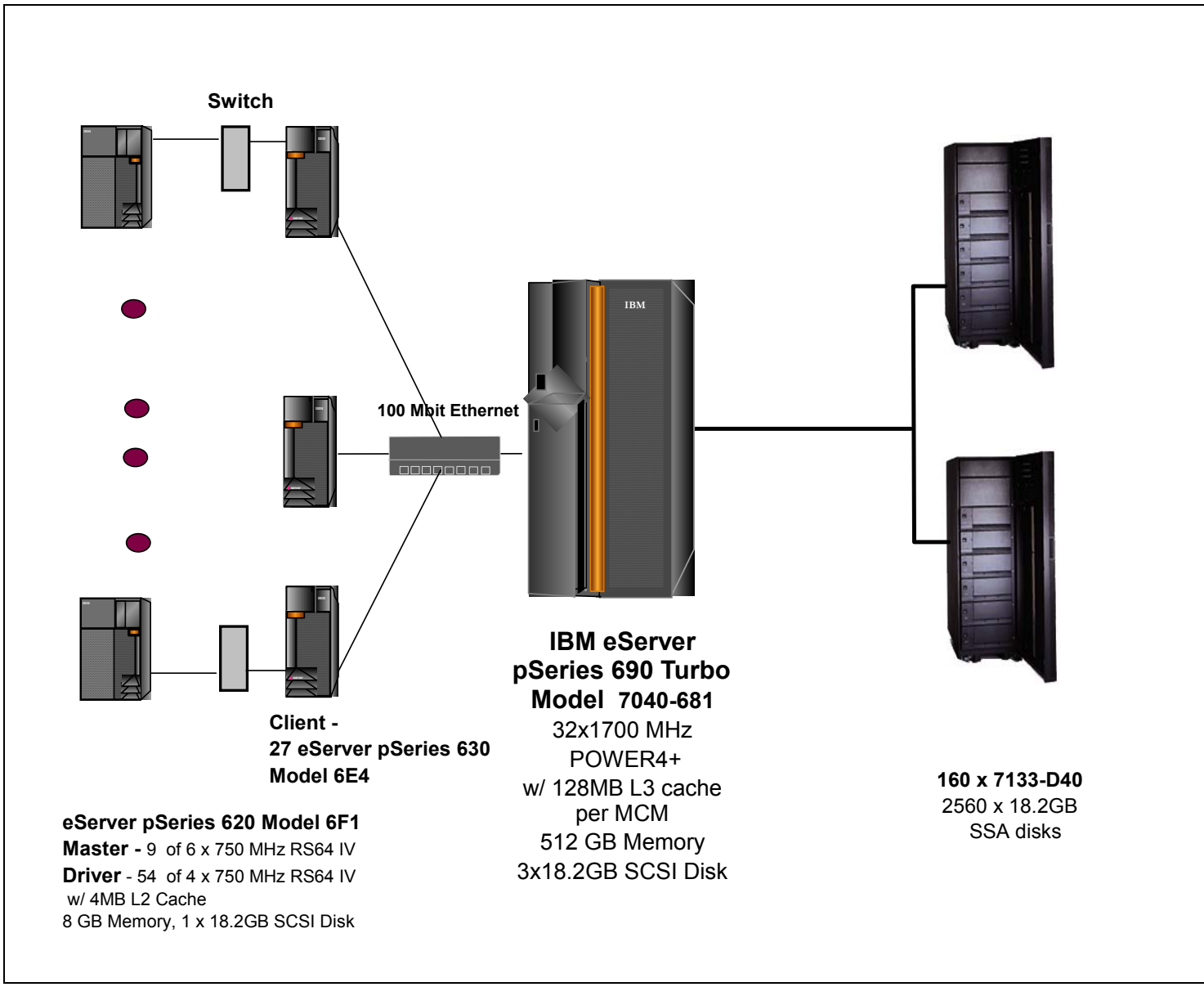
Appendix B contains the system, data base, and application parameters changed from their default values used in these TPC Benchmark™ C tests.

1.4 Configuration Diagrams

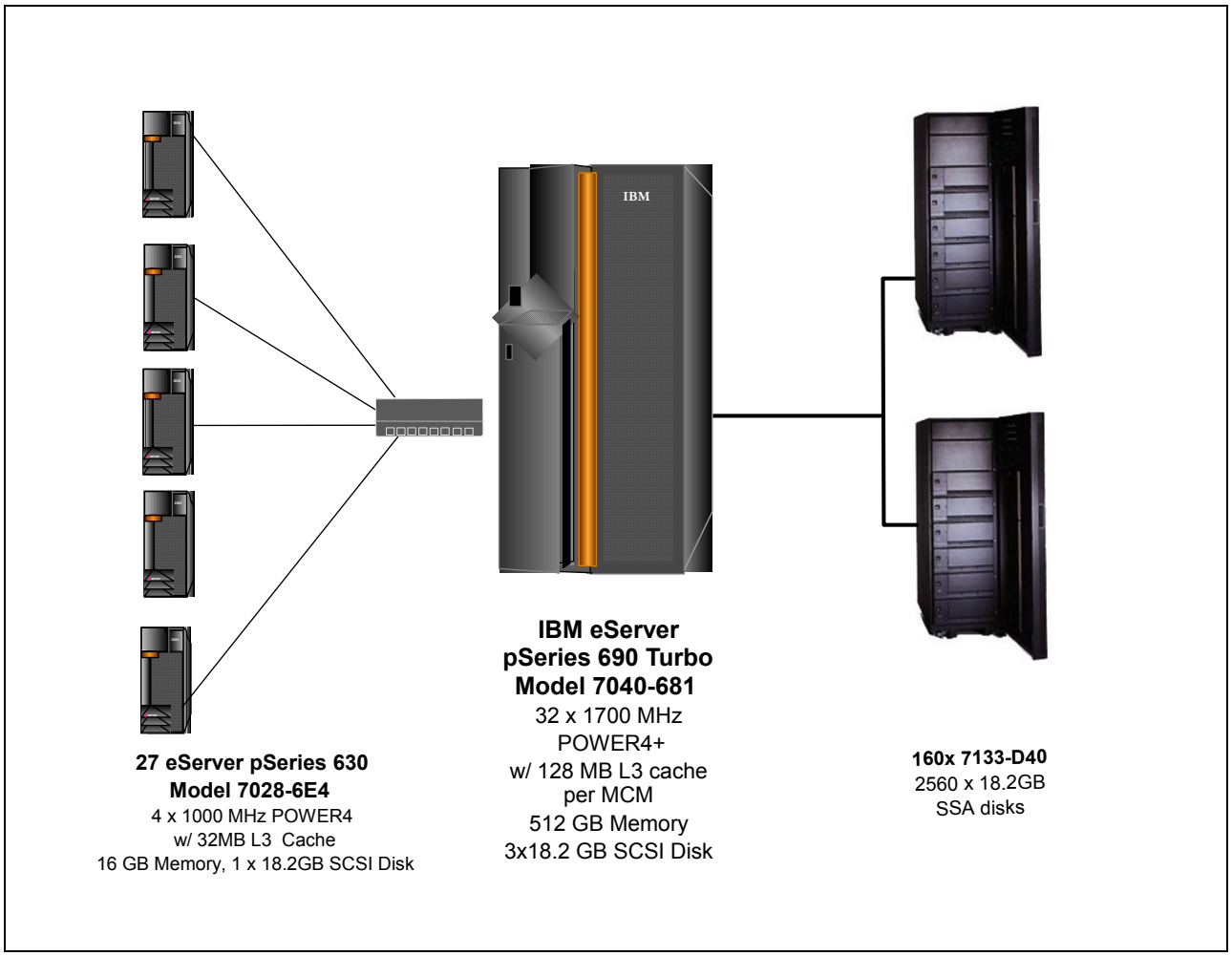
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- *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



The differences between the measured system and the priced system are as follows:

- The Server has 7 ethernet adapters that were not priced. These were used for the site network, file server, and extras. Netstat showed no network activity to these adaptors. Two of the installed SSA adaptors were not used.
- On the Clients, one disk was installed but not used. It has no filesystems or logical volumes defined on it.

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 DB2 UDB on the server disks according to the details provided in Appendix C.

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 DB2 UDB and priced as static space.

The insert and delete functions were verified by the auditor. In addition, the auditor verified that the primary key for each database table could be updated outside the range of its initial partition.

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.

The tables Customer, Order Line, and Stock were horizontally partitioned into multiple tables by warehouse ID. Each table partition had 1900 warehouses. Other tables were not partitioned. No tables were replicated. A view was created over all partitioned tables to provide full transparency of data manipulation.

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 seeds for each user were captured and verified by the auditor to be unique. In addition, the contents of the database were systematically searched, and randomly sampled by the auditor for patterns that would indicate the random number generator had affected any kind of a discernible pattern; none were found.

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 eServer pSeries 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 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 shows 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 to servers using the same TUXEDO mechanism that other transactions used. The only difference was that the call was asynchronous, i.e. control would return to the TUXEDO client process immediately and the deferred delivery part would complete asynchronously. 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	85.01%
Percentage of Remote transactions	14.99%
Non-Primary Key Access	
Percentage of Payment using C_LAST	59.99%
Percentage of Order-Status using C_LAST	60.01%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.91%
Payment	43.02%
Order-Status	4.02%
Delivery	4.01%
Stock-Level	4.01%

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

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 implemented with a Perl script that allowed the transaction to be rolled back rather than committed.
2. The balance, BALANCE_3, was retrieved from the Customer table for the same Customer, District, and Warehouse combination used in the completed Payment transaction Atomicity test.
3. The Payment transaction was executed for the Customer, District and Warehouse used in step 2. Rather than commit the transaction, the transaction was rolled back.
4. The balance, BALANCE_4 was retrieved again for the Customer, District, and Warehouse combination used in step 2. It was verified that BALANCE_4 was equal to BALANCE_3, demonstrating that there were no remaining effects of the rolled back Payment transaction.

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.

The specification defines 12 consistency conditions of which the following four are required to be explicitly demonstrated:

1. The sum of balances (d_ytd) for all Districts within a specific Warehouse is equal to the balance (w_ytd) of that Warehouse.

2. For each District within a Warehouse, the next available Order ID (d_next_o_id) minus one is equal to the most recent Order ID [max(o_id)] for the Order table associated with the preceeding District and Warehouse. Additionally, that same relationship exists for the most recent Order ID [max(o_id)] for the New Order table associated with the same District and Warehouse. Those relationships can be illustrated as follows:

   d_next_o_id - 1 = max(o_id) = max(no_o_id)

   where (d_w_id = o_w_id = no_w_id) and (d_id = o_d_id = no_d_id)

3. For each District within a Warehouse, the value of the most recent Order ID [max(no_o_id)] minus the first Order ID [min(no_o_id)] plus one, for the New Order table associated with the District and Warehouse equals the number of rows in that New Order table. That relationship can be illustrated as follows:

   max(no_o_id) - min(no_o_id) + 1 = number of rows in New Order for the Warehouse/District

4. For each District within a Warehouse, the sum of Order Line counts [sum(o_ol_cnt)] for the Order table associated with the District equals the number of rows in the Order Line table associated with the same District. That relationship can be illustrated as follows:

   sum(o_ol_cnt) = number of rows in the Order Line table for the Warehouse/District

An RTE driven run was executed against a freshly loaded database. After the run the 4 consistency conditions defined above were tested using a script to issue queries to the database. All queries showed 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.

The benchmark specification defines nine tests to demonstrate the property of transaction isolation. The tests, described in Clauses 3.4.2.1 - 3.4.2.9 were all successfully executed using a series of scripts. Case A was observed during the execution of Isolation Tests 7-9.

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 failure of Durable Fast Write Cache on SSA Adapter for Redo Logs.

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 full load test was started and allowed to run for over 10 minutes.
3. One of the disks containing the transaction log data was removed. Since the log was on a RAID5 disk, DB2 continued to process the transactions successfully.
4. The test continued for at least another 5 minutes.
5. Test B of the *Durable Fast Write Cache on SSA Adapter* was conducted. The system had already reached steady state when the fast write cache battery was failed using a toggle switch. The system recorded the failure, flushed out its VRAM contents and quit using write ahead caching. Error notices were posted into the system error log.
6. The run continued without the cache.

7. The system was subsequently shut down.
8. The disk from step 3 was replaced.
9. The system was powered back on and and DB2 allowed to recover.
10. Step 1 is performed returning the value for SUM_2. It was verified that SUM_2 was greater than SUM_1 plus the completed New_Order transactions recorded by the RTE. The additional transactions found in the database being attributed to in-flight activity at the time of the failure.

Failure of Durable Medium containing TPC-C database tables.

The following steps were successfully performed to demonstrate Durability against the failure of a disk unit with database tables:

1. The contents of the database was backed up in full.
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 with 22% of the full load.
4. A disk containing the TPCC table was removed.
5. The run was left to complete, recording many errors due to the disk failure.
6. The disk from step 4 was powered back on and the full database was restored from the backup copy in step 1.
7. DB2 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 greater than SUM_1 plus the completed New_Order transactions recorded by the RTE. The additional transactions found in the database being attributed to in-flight activity at the time of the failure.
9. Consistency condition 3 was verified.

Instantaneous Interruption and Memory Failure

The following steps were conducted on a fully scaled database:

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 full load test was started and allowed to run for over 10 minutes
3. Processing was brutally interrupted with an instantaneous halt.
4. The system was powered back on and DB2 recovered.
5. Step 1 was performed returning SUM_2. It was verified that SUM_2 was greater than SUM_1 plus the completed New_Order transactions recorded by the RTE. The additional transactions found in the database being attributed to in-flight activity at the time of the failure
6. Consistency condition 3 was verified.

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

These tests were conducted to verify the durability of the redundant Fast Write Cache. 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:

1. A) Failure of the adapter. This was executed along with the system crash test.
2. B) Failure of the onboard battery. This was executed along with the log disk failure test.

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	53,200
District	532,000
Customer	1,596,000,000
History	1,596,000,000
Order	1,596,000,000
New Order	478,800,000
Order Line	15,960,647,808
Stock	5,320,000,000
Item	100,000

5.2 Distribution of Tables and Logs

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

All disks are 18 GB SSA drives. All adaptors use RAID0 except adaptors SSA14 and SSA20 which use RAID5 for the log. There are 2 SSA adaptors for log, each adaptor has 6 hdisks, and each hdisk has 8 pdisks, for a total of 96 log disks. There are 28 SSA adaptors for data, each adaptor has 8 hdisks, and each hdisk has 11 pdisks, for a total of 2464 data disks.

Each SSA adaptor for data has 12 Logical Volumes (LV) stripped across 4 hdisks, and another set of 12 LVs stripped across the next 4 hdisks.

The LV name uses the following convention: D1FaaVbttt##, where

aa = rack number
b = Volume Group (VG) 1 or 2
ttt= table name
= the partition number of the table.

e.g. D1F01V1STK01 is the partitioned table of STK01 (stock1) on VG1 and rack 1

Here's the disk mapping for data:

```
Adaptor  Hdisk Pdisk

ssa0

        hdisk2 {0 4 6 9 10 11 12 13 21 22 23}
        hdisk3 {24 25 26 28 29 30 32 33 35 36 37}
        hdisk4 {38 39 40 41 42 43 44 45 46 47 48}
        hdisk5 {50 51 53 54 55 56 57 59 60 62 68}
        hdisk6 {1 2 3 5 7 8 14 15 16 17 18}
        hdisk7 {19 20 27 31 34 49 52 58 61 63 64}
        hdisk8 {65 66 67 70 72 73 74 75 76 77 78}
        hdisk9 {79 80 81 82 83 84 85 86 87 88 89}

        D1F01V1ITEM  D1F01V1WARE  D1F01V1DIST  D1F01V1NORI  D1F01V1CSTI01
        D1F01V1ORL01  D1F01V1STK01  D1F01V1CST01  D1F01V1ORDI  D1F01V1ORD
        D1F01V1NOR  D1F01V1HIST   D1F01V2ITEM  D1F01V2WARE  D1F01V2DIST
        D1F01V2NORI  D1F01V2CSTI01  D1F01V2ORL01  D1F01V2STK01  D1F01V2CST01
        D1F01V2ORDI  D1F01V2ORD  D1F01V2NOR  D1F01V2HIST

        D1tpccdblv  (database directory)

ssa1

        hdisk10 {190 96 97 100 101 105 108 109 110 112 114}
        hdisk106 {115 117 119 120 129 133 134 135 136 137 138}
        hdisk107 {142 143 146 147 160 162 163 164 165 166 167}
        hdisk108 {168 169 170 171 172 173 174 176 177 178 180}
        hdisk109 {98 99 102 103 104 106 107 111 113 116 118}
        hdisk110 {121 122 123 124 125 126 127 128 130 131 132}
        hdisk111 {139 140 141 144 145 148 149 150 151 152 153}
        hdisk112 {154 155 156 157 158 159 161 175 179 181 182}

        D1F02V1ITEM  D1F02V1WARE  D1F02V1DIST  D1F02V1NORI  D1F02V1CSTI02
        D1F02V1ORL02  D1F02V1STK02  D1F02V1CST02  D1F02V1ORDI  D1F02V1ORD
        D1F02V1NOR  D1F02V1HIST  D1F02V2ITEM  D1F02V2WARE  D1F02V2DIST
        D1F02V2NORI  D1F02V2CSTI02  D1F02V2ORL02  D1F02V2STK02  D1F02V2CST02
        D1F02V2ORDI  D1F02V2ORD  D1F02V2NOR  D1F02V2HIST

ssa2

        hdisk19 {287 192 193 194 195 200 205 208 209 211 215}
        hdisk39 {221 224 228 230 231 232 241 243 244 246 247}
        hdisk40 {249 252 253 254 257 258 261 262 263 266 267}
        hdisk41 {268 269 270 273 274 275 276 277 278 280 281}
        hdisk42 {196 197 198 199 201 202 203 204 206 207 210}
        hdisk43 {212 213 214 216 217 218 219 220 222 223 225}
        hdisk48 {226 227 229 233 234 235 236 237 238 239 240}
        hdisk49 {242 245 248 250 251 255 256 259 260 264 265}
```

```
        D1F03V1ITEM  D1F03V1WARE  D1F03V1DIST  D1F03V1NORI  D1F03V1CSTI03
        D1F03V1ORL03  D1F03V1STK03  D1F03V1CST03  D1F03V1ORDI  D1F03V1ORD
        D1F03V1NOR  D1F03V1HIST  D1F03V2ITEM  D1F03V2WARE  D1F03V2DIST
        D1F03V2NORI  D1F03V2CSTI03  D1F03V2ORL03  D1F03V2STK03  D1F03V2CST03

        D1F03V2ORDI  D1F03V2ORD  D1F03V2NOR  D1F03V2HIST

ssa3

        hdisk160 {288 290 291 292 296 297 298 299 300 301 304}
        hdisk161 {308 310 316 318 322 323 326 327 328 334 336}
        hdisk162 {337 338 341 344 346 348 353 355 356 358 360}
        hdisk163 {362 364 366 369 370 371 372 373 374 375 377}
        hdisk164 {289 293 294 295 302 303 305 306 307 309 311}
        hdisk165 {312 313 314 315 317 319 320 321 324 325 329}
        hdisk166 {330 331 332 333 339 340 342 343 345 347 368}
        hdisk167 {349 350 351 352 354 357 359 361 363 365 367}

        D1F04V1ITEM  D1F04V1WARE  D1F04V1DIST  D1F04V1NORI  D1F04V1CSTI04
        D1F04V1ORL04  D1F04V1STK04  D1F04V1CST04  D1F04V1ORDI  D1F04V1ORD
        D1F04V1NOR  D1F04V1HIST  D1F04V2ITEM  D1F04V2WARE  D1F04V2DIST
        D1F04V2NORI  D1F04V2CSTI04  D1F04V2ORL04  D1F04V2STK04  D1F04V2CST04
        D1F04V2ORDI  D1F04V2ORD  D1F04V2NOR  D1F04V2HIST

ssa4

        hdisk176 {384 385 391 394 395 396 400 401 402 403 404}
        hdisk177 {410 413 414 415 417 418 421 423 424 425 428}
        hdisk178 {430 431 436 440 446 447 448 452 453 454 455}
        hdisk179 {456 457 458 459 461 462 464 465 471 472 473}
        hdisk180 {386 387 388 389 390 392 393 397 398 399 405}
        hdisk181 {406 407 408 409 411 412 416 419 420 422 426}
        hdisk182 {427 429 432 433 434 435 437 438 439 441 442}
        hdisk183 {443 444 445 449 450 451 460 463 466 467 468}

        D1F05V1ITEM  D1F05V1WARE  D1F05V1DIST  D1F05V1NORI  D1F05V1CSTI05
        D1F05V1ORL05  D1F05V1STK05  D1F05V1CST05  D1F05V1ORDI  D1F05V1ORD
        D1F05V1NOR  D1F05V1HIST  D1F05V2ITEM  D1F05V2WARE  D1F05V2DIST
        D1F05V2NORI  D1F05V2CSTI05  D1F05V2ORL05  D1F05V2STK05  D1F05V2CST05
        D1F05V2ORDI  D1F05V2ORD  D1F05V2NOR  D1F05V2HIST

ssa5

        hdisk44 {480 481 482 484 485 490 491 493 566 568 569}
        hdisk45 {494 495 497 498 500 505 507 508 509 512 518}
        hdisk46 {519 523 525 526 528 529 533 542 544 546 548}
        hdisk47 {553 554 555 557 558 559 560 561 562 564 565}
        hdisk184 {570 575 483 486 487 488 489 496 499 501 502}
        hdisk185 {503 504 506 510 511 513 514 515 516 517 520}
        hdisk186 {521 522 524 527 530 531 532 534 535 536 537}
        hdisk187 {538 539 540 541 543 545 547 556 550 551 552}

        D1F06V1ITEM  D1F06V1WARE  D1F06V1DIST  D1F06V1NORI  D1F06V1CSTI06
        D1F06V1ORL06  D1F06V1STK06  D1F06V1CST06  D1F06V1ORDI  D1F06V1ORD
        D1F06V1NOR  D1F06V1HIST  D1F06V2ITEM  D1F06V2WARE  D1F06V2DIST
        D1F06V2NORI  D1F06V2CSTI06  D1F06V2ORL06  D1F06V2STK06  D1F06V2CST06
        D1F06V2ORDI  D1F06V2ORD  D1F06V2NOR  D1F06V2HIST

ssa6

        hdisk188 {576 583 584 586 590 591 592 593 596 597 598}
        hdisk189 {599 602 604 606 607 608 611 612 613 618 621}
        hdisk190 {622 626 628 629 630 631 632 633 634 635 636}
        hdisk191 {638 639 640 641 650 652 654 655 656 657 666}
        hdisk192 {577 578 579 580 581 582 585 587 588 589 594}
        hdisk193 {595 600 601 603 605 609 610 614 615 616 617}
        hdisk194 {619 620 623 624 625 627 637 642 643 644 645}
        hdisk195 {646 647 648 649 651 653 658 659 660 661 662}

        D1F07V1ITEM  D1F07V1WARE  D1F07V1DIST  D1F07V1NORI  D1F07V1CSTI07
        D1F07V1ORL07  D1F07V1STK07  D1F07V1CST07  D1F07V1ORDI  D1F07V1ORD
        D1F07V1NOR  D1F07V1HIST  D1F07V2ITEM  D1F07V2WARE  D1F07V2DIST
        D1F07V2NORI  D1F07V2CSTI07  D1F07V2ORL07  D1F07V2STK07  D1F07V2CST07
        D1F07V2ORDI  D1F07V2ORD  D1F07V2NOR  D1F07V2HIST

Ssa7
```

```
        hdisk196 {684 685 686 687 690 691 692 693 697 700 701}
        hdisk197 {702 705 709 714 715 717 718 719 720 721 722}
        hdisk198 {723 724 725 728 732 734 735 740 741 742 743}
        hdisk199 {744 745 746 747 748 749 751 752 754 757 759}
        hdisk200 {672 673 674 675 676 677 678 679 680 681 682}
        hdisk201 {683 688 689 694 695 696 698 699 703 704 706}
        hdisk202 {707 708 710 711 712 713 716 726 727 729 730}
        hdisk203 {731 733 736 737 738 739 750 753 755 756 758}

        D1F08V1ITEM  D1F08V1WARE  D1F08V1DIST  D1F08V1NORI  D1F08V1CSTI08
        D1F08V1ORL08  D1F08V1STK08  D1F08V1CST08  D1F08V1ORDI  D1F08V1ORD
        D1F08V1NOR  D1F08V1HIST  D1F08V2ITEM  D1F08V2WARE  D1F08V2DIST
        D1F08V2NORI  D1F08V2CSTI08  D1F08V2ORL08  D1F08V2STK08  D1F08V2CST08
        D1F08V2ORDI  D1F08V2ORD  D1F08V2NOR  D1F08V2HIST

ssa8

        hdisk72 {862 768 770 772 773 774 778 780 785 787 788}
        hdisk204 {791 797 799 801 804 807 809 812 813 814 815}
        hdisk205 {816 819 820 821 826 828 831 832 833 842 843}
        hdisk206 {844 845 847 848 849 851 852 853 856 857 858}
        hdisk207 {769 771 775 776 777 779 781 782 783 784 786}
        hdisk208 {789 790 792 793 794 795 796 798 800 802 803}
        hdisk209 {805 806 808 810 811 817 818 824 825 827 829}
        hdisk210 {830 834 835 836 837 838 839 840 841 846 850}

        D1F09V1ITEM  D1F09V1WARE  D1F09V1DIST  D1F09V1NORI  D1F09V1CSTI09
        D1F09V1ORL09  D1F09V1STK09  D1F09V1CST09  D1F09V1ORDI  D1F09V1ORD
        D1F09V1NOR  D1F09V1HIST  D1F09V2ITEM  D1F09V2WARE  D1F09V2DIST
        D1F09V2NORI  D1F09V2CSTI09  D1F09V2ORL09  D1F09V2STK09  D1F09V2CST09
        D1F09V2ORDI  D1F09V2ORD  D1F09V2NOR  D1F09V2HIST

ssa9

        hdisk81 {873 870 871 878 879 883 885 886 887 888 890}
        hdisk211 {891 895 896 900 904 905 907 908 910 911 913}
        hdisk212 {915 917 918 920 921 922 923 925 926 927 929}
        hdisk213 {930 934 935 937 939 940 941 943 946 949 952}
        hdisk214 {864 865 866 867 868 869 872 874 875 876 877}
        hdisk215 {880 881 882 884 889 892 893 894 897 898 899}
        hdisk216 {901 902 903 906 909 912 914 916 919 924 928}
        hdisk217 {931 932 933 936 938 942 944 947 948 950 951}

        D1F10V1ITEM  D1F10V1WARE  D1F10V1DIST  D1F10V1NORI  D1F10V1CSTI10
        D1F10V1ORL10  D1F10V1STK10  D1F10V1CST10  D1F10V1ORDI  D1F10V1ORD
        D1F10V1NOR  D1F10V1HIST  D1F10V2ITEM  D1F10V2WARE  D1F10V2DIST
        D1F10V2NORI  D1F10V2CSTI10  D1F10V2ORL10  D1F10V2STK10  D1F10V2CST10
        D1F10V2ORDI  D1F10V2ORD  D1F10V2NOR  D1F10V2HIST

ssa10

        hdisk113 {960 961 962 963 964 967 968 969 970 979 982}
        hdisk115 {983 984 985 989 994 995 996 998 999 1000 1001}
        hdisk116 {1004 1005 1008 1009 1010 1011 1012 1013 1014 1015 1016}
        hdisk117 {1017 1018 1019 1020 1021 1024 1026 1033 1034 1035 1036}
        hdisk118 {965 966 971 972 973 974 975 976 977 978 980}
        hdisk119 {981 986 987 988 990 991 992 993 997 1002 1003}
        hdisk120 {1006 1007 1022 1023 1025 1027 1028 1029 1030 1031 1032}
        hdisk121 {1037 1039 1041 1042 1043 1044 1045 1046 1047 1048 1049}

        D1F11V1ITEM  D1F11V1WARE  D1F11V1DIST  D1F11V1NORI  D1F11V1CSTI11
        D1F11V1ORL11  D1F11V1STK11  D1F11V1CST11  D1F11V1ORDI  D1F11V1ORD
        D1F11V1NOR  D1F11V1HIST  D1F11V2ITEM  D1F11V2WARE  D1F11V2DIST
        D1F11V2NORI  D1F11V2CSTI11  D1F11V2ORL11  D1F11V2STK11  D1F11V2CST11
        D1F11V2ORDI  D1F11V2ORD  D1F11V2NOR  D1F11V2HIST

ssa11

        hdisk122 {1059 1060 1061 1065 1066 1067 1069 1070 1071 1072 1076}
        hdisk123 {1078 1081 1082 1083 1084 1091 1093 1095 1099 1100 1101}
        hdisk124 {1103 1108 1110 1113 1114 1115 1116 1117 1118 1119 1126}
        hdisk125 {1128 1129 1130 1131 1133 1134 1135 1136 1137 1138 1139}
        hdisk126 {1056 1057 1058 1062 1063 1064 1068 1073 1074 1075 1077}
        hdisk127 {1079 1080 1085 1086 1087 1088 1089 1090 1092 1094 1096}
        hdisk128 {1097 1098 1102 1104 1105 1106 1107 1109 1111 1112 1120}
        hdisk130 {1121 1122 1123 1124 1125 1127 1132 1140 1141 1144 1145}
```

```
        D1F12V1ITEM  D1F12V1WARE  D1F12V1DIST  D1F12V1NORI  D1F12V1CSTI12
        D1F12V1ORL12  D1F12V1STK12  D1F12V1CST12  D1F12V1ORDI  D1F12V1ORD
        D1F12V1NOR  D1F12V1HIST  D1F12V2ITEM  D1F12V2WARE  D1F12V2DIST
        D1F12V2NORI  D1F12V2CSTI12  D1F12V2ORL12  D1F12V2STK12  D1F12V2CST12
        D1F12V2ORDI  D1F12V2ORD  D1F12V2NOR  D1F12V2HIST

ssa12

        hdisk132 {1153 1155 1161 1162 1163 1164 1165 1166 1168 1171 1172}
        hdisk133 {1173 1174 1175 1176 1177 1178 1182 1183 1184 1185 1186}
        hdisk134 {1188 1189 1190 1192 1193 1195 1199 1201 1205 1208 1210}
        hdisk135 {1211 1214 1215 1216 1218 1219 1220 1221 1223 1230 1231}
        hdisk136 {1180 1181 1187 1191 1194 1196 1197 1198 1200 1202 1203}
        hdisk137 {1204 1206 1207 1209 1212 1213 1217 1222 1224 1225 1226}
        hdisk138 {1227 1228 1229 1233 1234 1235 1236 1237 1238 1239 1240}
        hdisk218 {1152 1158 1159 1160 1167 1169 1170 1179 1241 1242 1244}

        D1F13V1ITEM  D1F13V1WARE  D1F13V1DIST  D1F13V1NORI  D1F13V1CSTI13
        D1F13V1ORL13  D1F13V1STK13  D1F13V1CST13  D1F13V1ORDI  D1F13V1ORD
        D1F13V1NOR  D1F13V1HIST  D1F13V2ITEM  D1F13V2WARE  D1F13V2DIST
        D1F13V2NORI  D1F13V2CSTI13  D1F13V2ORL13  D1F13V2STK13  D1F13V2CST13
        D1F13V2ORDI  D1F13V2ORD  D1F13V2NOR  D1F13V2HIST

ssa13

        hdisk114 {1248 1253 1259 1261 1262 1263 1264 1265 1266 1267 1279}
        hdisk231 {1281 1282 1283 1284 1285 1286 1287 1288 1289 1294 1295}
        hdisk232 {1296 1298 1299 1300 1301 1302 1304 1305 1306 1307 1309}
        hdisk233 {1310 1311 1312 1314 1315 1318 1319 1320 1321 1322 1326}
        hdisk234 {1343 1249 1250 1251 1252 1254 1255 1256 1257 1258 1260}
        hdisk262 {1269 1270 1271 1272 1273 1274 1275 1276 1277 1342}
        hdisk236 {1280 1290 1291 1292 1293 1297 1303 1308 1313 1316 1317}
        hdisk237 {1323 1324 1325 1328 1329 1330 1332 1333 1335 1336 1338}



        D1F14V1ITEM  D1F14V1WARE  D1F14V1DIST  D1F14V1NORI  D1F14V1CSTI14
        D1F14V1ORL14  D1F14V1STK14  D1F14V1CST14  D1F14V1ORDI  D1F14V1ORD
        D1F14V1NOR  D1F14V1HIST  D1F14V2ITEM  D1F14V2WARE  D1F14V2DIST
        D1F14V2NORI  D1F14V2CSTI14  D1F14V2ORL14  D1F14V2STK14  D1F14V2CST14
        D1F14V2ORDI  D1F14V2ORD  D1F14V2NOR  D1F14V2HIST

ssa15

        hdisk129 {1392 1394 1398 1400 1401 1402 1405 1406 1408 1409 1410}
        hdisk238 {1411 1412 1415 1424 1429 1432 1433 1435 1436 1439 1440}
        hdisk239 {1444 1445 1446 1449 1450 1452 1453 1454 1455 1456 1457}
        hdisk240 {1458 1461 1464 1466 1468 1471 1473 1476 1478 1480 1482}
        hdisk241 {1487 1393 1395 1396 1397 1399 1403 1404 1407 1413 1414}
        hdisk242 {1416 1417 1418 1419 1420 1421 1422 1423 1425 1426 1427}
        hdisk243 {1428 1430 1431 1434 1437 1438 1441 1442 1443 1447 1448}
        hdisk244 {1451 1459 1460 1462 1463 1465 1467 1469 1470 1472 1474}

        D1F15V1ITEM  D1F15V1WARE  D1F15V1DIST  D1F15V1NORI  D1F15V1CSTI15
        D1F15V1ORL15  D1F15V1STK15  D1F15V1CST15  D1F15V1ORDI  D1F15V1ORD
        D1F15V1NOR  D1F15V1HIST  D1F15V2ITEM  D1F15V2WARE  D1F15V2DIST
        D1F15V2NORI  D1F15V2CSTI15  D1F15V2ORL15  D1F15V2STK15  D1F15V2CST15
        D1F15V2ORDI  D1F15V2ORD  D1F15V2NOR  D1F15V2HIST

ssa16

        hdisk11 {1490 1491 1492 1493 1496 1498 1506 1507 1509 1510 1511}
        hdisk12 {1513 1514 1515 1516 1521 1523 1524 1530 1531 1535 1536}
        hdisk13 {1537 1539 1540 1541 1543 1544 1546 1547 1549 1550 1552}
        hdisk139 {1557 1558 1560 1561 1563 1564 1567 1569 1574 1576 1577}
        hdisk140 {1529 1532 1533 1488 1489 1494 1495 1497 1499 1500 1501}
        hdisk141 {1502 1503 1504 1505 1512 1517 1518 1519 1520 1522 1525}
        hdisk142 {1526 1527 1528 1534 1538 1542 1545 1548 1551 1553 1554}
        hdisk143 {1555 1556 1559 1562 1565 1566 1568 1570 1571 1572 1573}

        D1F16V1ITEM  D1F16V1WARE  D1F16V1DIST  D1F16V1NORI  D1F16V1CSTI16
        D1F16V1ORL16  D1F16V1STK16  D1F16V1CST16  D1F16V1ORDI  D1F16V1ORD
        D1F16V1NOR  D1F16V1HIST  D1F16V2ITEM  D1F16V2WARE  D1F16V2DIST
        D1F16V2NORI  D1F16V2CSTI16  D1F16V2ORL16  D1F16V2STK16  D1F16V2CST16
        D1F16V2ORDI  D1F16V2ORD  D1F16V2NOR  D1F16V2HIST

ssa17
```

```
        hdisk14 {1596 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607}
        hdisk15 {1608 1609 1612 1613 1622 1623 1625 1630 1631 1634 1640}
        hdisk16 {1641 1642 1643 1645 1646 1647 1649 1650 1651 1653 1654}
        hdisk17 {1656 1657 1658 1659 1660 1662 1663 1664 1665 1667 1673}
        hdisk18 {1675 1666 1584 1585 1586 1587 1588 1589 1590 1591 1592}
        hdisk20 {1593 1594 1595 1597 1610 1611 1614 1615 1616 1617 1618}
        hdisk21 {1619 1620 1621 1624 1626 1627 1628 1629 1632 1633 1635}
        hdisk22 {1636 1637 1638 1639 1644 1648 1652 1655 1661 1668 1669}

        D1F17V1ITEM  D1F17V1WARE  D1F17V1DIST  D1F17V1NORI  D1F17V1CSTI17
        D1F17V1ORL17  D1F17V1STK17  D1F17V1CST17  D1F17V1ORDI  D1F17V1ORD
        D1F17V1NOR  D1F17V1HIST  D1F17V2ITEM  D1F17V2WARE  D1F17V2DIST
        D1F17V2NORI  D1F17V2CSTI17  D1F17V2ORL17  D1F17V2STK17  D1F17V2CST17
        D1F17V2ORDI  D1F17V2ORD  D1F17V2NOR  D1F17V2HIST

ssa18

        hdisk23 {1680 1682 1683 1684 1690 1692 1693 1694 1696 1698 1700}
        hdisk24 {1701 1702 1703 1704 1706 1707 1708 1710 1712 1715 1716}
        hdisk25 {1717 1718 1720 1721 1722 1723 1724 1725 1727 1729 1732}
        hdisk26 {1733 1735 1736 1737 1738 1740 1747 1749 1758 1762 1764}
        hdisk27 {1681 1685 1686 1687 1688 1689 1691 1695 1697 1699 1705}
        hdisk28 {1709 1711 1713 1714 1719 1726 1728 1730 1731 1734 1739}
        hdisk29 {1741 1742 1743 1744 1745 1746 1748 1750 1751 1752 1753}
        hdisk30 {1754 1755 1756 1757 1759 1760 1761 1763 1765 1768 1769}

        D1F18V1ITEM  D1F18V1WARE  D1F18V1DIST  D1F18V1NORI  D1F18V1CSTI18
        D1F18V1ORL18  D1F18V1STK18  D1F18V1CST18  D1F18V1ORDI  D1F18V1ORD
        D1F18V1NOR  D1F18V1HIST  D1F18V2ITEM  D1F18V2WARE  D1F18V2DIST
        D1F18V2NORI  D1F18V2CSTI18  D1F18V2ORL18  D1F18V2STK18  D1F18V2CST18
        D1F18V2ORDI  D1F18V2ORD  D1F18V2NOR  D1F18V2HIST

ssa19

        hdisk31 {1780 1781 1782 1783 1784 1785 1786 1787 1790 1791 1792}
        hdisk32 {1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803}
        hdisk33 {1804 1805 1806 1807 1808 1809 1811 1816 1817 1822 1864}
        hdisk34 {1823 1829 1835 1836 1840 1843 1844 1847 1848 1850 1854}
        hdisk35 {1776 1777 1778 1779 1788 1789 1810 1812 1813 1815 1818}
        hdisk36 {1819 1820 1821 1824 1825 1826 1827 1828 1830 1831 1832}
        hdisk37 {1833 1834 1837 1838 1839 1841 1842 1845 1846 1849 1851}
        hdisk38 {1852 1853 1855 1856 1857 1858 1859 1860 1861 1862 1863}


        D1F19V1ITEM  D1F19V1WARE  D1F19V1DIST  D1F19V1NORI  D1F19V1CSTI19
        D1F19V1ORL19  D1F19V1STK19  D1F19V1CST19  D1F19V1ORDI  D1F19V1ORD
        D1F19V1NOR  D1F19V1HIST  D1F19V2ITEM  D1F19V2WARE  D1F19V2DIST
        D1F19V2NORI  D1F19V2CSTI19  D1F19V2ORL19  D1F19V2STK19  D1F19V2CST19
        D1F19V2ORDI  D1F19V2ORD  D1F19V2NOR  D1F19V2HIST

ssa21

        hdisk50 {1922 1923 1924 1925 1926 1927 1928 1930 1931 1932 1934}
        hdisk51 {1937 1938 1939 1941 1947 1948 1949 1950 1951 1957 1968}
        hdisk52 {1969 1970 1972 1973 1974 1975 1976 1977 1978 1979 1981}
        hdisk53 {1982 1985 1987 1988 1996 2000 2001 2003 2005 2007 2008}
        hdisk54 {1920 1921 1929 1933 1935 1936 1940 1942 1943 1944 1945}
        hdisk55 {1946 1952 1953 1954 1955 1956 1958 1959 1960 1961 1962}
        hdisk56 {1963 1964 1965 1966 1967 1971 1980 1983 1984 1986 1989}
        hdisk57 {1990 1991 1992 1993 1994 1995 1997 1998 1999 2002 2004}

        D1F20V1ITEM  D1F20V1WARE  D1F20V1DIST  D1F20V1NORI  D1F20V1CSTI20
        D1F20V1ORL20  D1F20V1STK20  D1F20V1CST20  D1F20V1ORDI  D1F20V1ORD
        D1F20V1NOR  D1F20V1HIST  D1F20V2ITEM  D1F20V2WARE  D1F20V2DIST
        D1F20V2NORI  D1F20V2CSTI20  D1F20V2ORL20  D1F20V2STK20  D1F20V2CST20
        D1F20V2ORDI  D1F20V2ORD  D1F20V2NOR  D1F20V2HIST

ssa22

        hdisk58 {2018 2028 2035 2041 2042 2043 2044 2045 2048 2049 2052}
        hdisk59 {2055 2056 2057 2058 2059 2061 2065 2066 2067 2068 2070}
        hdisk60 {2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2082}
        hdisk61 {2083 2090 2091 2096 2097 2098 2099 2100 2101 2102 2103}
        hdisk62 {2015 2016 2017 2019 2020 2021 2022 2023 2024 2025 2026}
        hdisk63 {2027 2029 2030 2031 2032 2033 2034 2036 2037 2038 2039}
        hdisk64 {2040 2046 2047 2050 2051 2053 2054 2060 2062 2063 2064}
        hdisk65 {2069 2081 2084 2085 2086 2087 2088 2089 2092 2093 2094}
```

```
        D1F21V1ITEM  D1F21V1WARE  D1F21V1DIST  D1F21V1NORI  D1F21V1CSTI21
        D1F21V1ORL21  D1F21V1STK21  D1F21V1CST21  D1F21V1ORDI  D1F21V1ORD
        D1F21V1NOR  D1F21V1HIST  D1F21V2ITEM  D1F21V2WARE  D1F21V2DIST
        D1F21V2NORI  D1F21V2CSTI21  D1F21V2ORL21  D1F21V2STK21  D1F21V2CST21
        D1F21V2ORDI  D1F21V2ORD  D1F21V2NOR  D1F21V2HIST

ssa23

        hdisk66 {2111 2113 2115 2116 2117 2118 2119 2121 2122 2124 2125}
        hdisk67 {2126 2127 2129 2130 2131 2133 2134 2135 2141 2143 2144}
        hdisk68 {2146 2150 2159 2169 2170 2174 2176 2178 2179 2181 2182}
        hdisk69 {2183 2184 2194 2195 2197 2198 2199 2201 2202 2203 2204}
        hdisk70 {2112 2114 2123 2128 2132 2136 2137 2138 2139 2140 2142}
        hdisk71 {2145 2147 2149 2151 2152 2153 2154 2155 2156 2157 2158}
        hdisk73 {2160 2161 2162 2163 2164 2165 2166 2167 2168 2171 2172}
        hdisk74 {2173 2175 2177 2180 2185 2186 2187 2188 2189 2190 2191}

        D1F22V1ITEM  D1F22V1WARE  D1F22V1DIST  D1F22V1NORI  D1F22V1CSTI22
        D1F22V1ORL22  D1F22V1STK22  D1F22V1CST22  D1F22V1ORDI  D1F22V1ORD
        D1F22V1NOR  D1F22V1HIST  D1F22V2ITEM  D1F22V2WARE  D1F22V2DIST
        D1F22V2NORI  D1F22V2CSTI22  D1F22V2ORL22  D1F22V2STK22  D1F22V2CST22
        D1F22V2ORDI  D1F22V2ORD  D1F22V2NOR  D1F22V2HIST

ssa24

        hdisk75 {2209 2210 2212 2214 2215 2218 2219 2222 2225 2226 2230}
        hdisk76 {2232 2233 2234 2235 2237 2238 2239 2240 2241 2244 2250}
        hdisk77 {2254 2255 2258 2259 2261 2296 2263 2269 2270 2271 2273}
        hdisk78 {2274 2275 2279 2285 2286 2289 2291 2292 2293 2294 2295}
        hdisk79 {2207 2208 2211 2213 2216 2217 2220 2221 2223 2224 2227}
        hdisk80 {2229 2231 2236 2242 2243 2245 2246 2247 2248 2249 2251}
        hdisk82 {2252 2253 2256 2257 2260 2264 2265 2266 2267 2268 2272}
        hdisk83 {2276 2277 2278 2280 2281 2282 2283 2284 2287 2288 2290}

        D1F23V1ITEM  D1F23V1WARE  D1F23V1DIST  D1F23V1NORI  D1F23V1CSTI23
        D1F23V1ORL23  D1F23V1STK23  D1F23V1CST23  D1F23V1ORDI  D1F23V1ORD
        D1F23V1NOR  D1F23V1HIST  D1F23V2ITEM  D1F23V2WARE  D1F23V2DIST
        D1F23V2NORI  D1F23V2CSTI23  D1F23V2ORL23  D1F23V2STK23  D1F23V2CST23
        D1F23V2ORDI  D1F23V2ORD  D1F23V2NOR  D1F23V2HIST

ssa25

        hdisk84 {2303 2304 2305 2306 2307 2309 2310 2313 2314 2319 2321}
        hdisk85 {2332 2333 2335 2336 2337 2338 2340 2341 2343 2348 2349}
        hdisk86 {2353 2354 2355 2356 2357 2362 2363 2364 2367 2369 2370}
        hdisk87 {2372 2374 2375 2378 2381 2382 2385 2387 2388 2389 2393}
        hdisk88 {2308 2312 2315 2316 2317 2318 2320 2322 2323 2324 2390}
        hdisk89 {2325 2326 2327 2328 2329 2330 2331 2334 2339 2342 2344}
        hdisk90 {2391 2346 2347 2350 2351 2352 2358 2359 2360 2361 2365}
        hdisk91 {2366 2368 2371 2373 2376 2377 2379 2380 2383 2384 2386}

        D1F24V1ITEM  D1F24V1WARE  D1F24V1DIST  D1F24V1NORI  D1F24V1CSTI24
        D1F24V1ORL24  D1F24V1STK24  D1F24V1CST24  D1F24V1ORDI  D1F24V1ORD
        D1F24V1NOR  D1F24V1HIST  D1F24V2ITEM  D1F24V2WARE  D1F24V2DIST
        D1F24V2NORI  D1F24V2CSTI24  D1F24V2ORL24  D1F24V2STK24  D1F24V2CST24
        D1F24V2ORDI  D1F24V2ORD  D1F24V2NOR  D1F24V2HIST

ssa26

        hdisk92 {2402 2407 2408 2412 2413 2414 2417 2420 2424 2425 2428}
        hdisk93 {2436 2439 2440 2441 2442 2444 2459 2462 2463 2464 2493}
        hdisk94 {2467 2468 2469 2470 2473 2474 2476 2478 2479 2489 2494}
        hdisk95 {2399 2400 2401 2403 2404 2405 2406 2409 2410 2411 2415}
        hdisk96 {2416 2418 2419 2421 2422 2423 2426 2427 2429 2430 2431}
        hdisk97 {2432 2433 2434 2435 2437 2438 2445 2446 2447 2448 2449}
        hdisk98 {2450 2451 2452 2453 2454 2455 2456 2457 2458 2460 2461}
        hdisk219 {2481 2482 2483 2484 2485 2486 2487 2488 2490 2491 2492}

        D1F25V1ITEM  D1F25V1WARE  D1F25V1DIST  D1F25V1NORI  D1F25V1CSTI25
        D1F25V1ORL25  D1F25V1STK25  D1F25V1CST25  D1F25V1ORDI  D1F25V1ORD
        D1F25V1NOR  D1F25V1HIST  D1F25V2ITEM  D1F25V2WARE  D1F25V2DIST
        D1F25V2NORI  D1F25V2CSTI25  D1F25V2ORL25  D1F25V2STK25  D1F25V2CST25
        D1F25V2ORDI  D1F25V2ORD  D1F25V2NOR  D1F25V2HIST

ssa27

        hdisk99 {2495 2496 2497 2498 2500 2501 2502 2504 2505 2506 2507}
```

```
        hdisk100 {2512 2513 2516 2517 2518 2521 2522 2523 2525 2526 2533}
        hdisk101 {2535 2536 2537 2538 2541 2542 2543 2555 2556 2557 2561}
        hdisk102 {2564 2566 2567 2568 2571 2572 2574 2575 2578 2581 2588}
        hdisk103 {2503 2508 2509 2510 2511 2514 2515 2519 2520 2524 2527}
        hdisk104 {2528 2529 2530 2531 2532 2534 2539 2540 2544 2545 2546}
        hdisk105 {2547 2548 2549 2550 2551 2552 2553 2558 2559 2560 2562}
        hdisk144 {2563 2565 2570 2573 2576 2577 2579 2580 2582 2583 2584}

        D1F26V1ITEM  D1F26V1WARE  D1F26V1DIST  D1F26V1NORI  D1F26V1CSTI26
        D1F26V1ORL26  D1F26V1STK26  D1F26V1CST26  D1F26V1ORDI  D1F26V1ORD
        D1F26V1NOR  D1F26V1HIST  D1F26V2ITEM  D1F26V2WARE  D1F26V2DIST
        D1F26V2NORI  D1F26V2CSTI26  D1F26V2ORL26  D1F26V2STK26  D1F26V2CST26
        D1F26V2ORDI  D1F26V2ORD  D1F26V2NOR  D1F26V2HIST

ssa28

        hdisk145 {2591 2593 2595 2596 2597 2598 2599 2602 2603 2617 2618}
        hdisk146 {2619 2620 2622 2623 2628 2686 2630 2631 2632 2633 2639}
        hdisk147 {2646 2648 2649 2652 2657 2658 2660 2661 2662 2664 2683}
        hdisk148 {2667 2668 2669 2671 2672 2674 2675 2676 2680 2681 2682}
        hdisk149 {2592 2594 2600 2601 2604 2605 2606 2607 2608 2609 2610}
        hdisk150 {2611 2612 2613 2614 2615 2616 2621 2624 2625 2626 2627}
        hdisk151 {2634 2635 2636 2637 2638 2640 2641 2642 2643 2644 2645}
        hdisk152 {2647 2651 2653 2654 2655 2659 2663 2665 2666 2670 2673}

        D1F27V1ITEM  D1F27V1WARE  D1F27V1DIST  D1F27V1NORI  D1F27V1CSTI27
        D1F27V1ORL27  D1F27V1STK27  D1F27V1CST27  D1F27V1ORDI  D1F27V1ORD
        D1F27V1NOR  D1F27V1HIST  D1F27V2ITEM  D1F27V2WARE  D1F27V2DIST
        D1F27V2NORI  D1F27V2CSTI27  D1F27V2ORL27  D1F27V2STK27  D1F27V2CST27
        D1F27V2ORDI  D1F27V2ORD  D1F27V2NOR  D1F27V2HIST

ssa29

        hdisk153 {2709 2710 2711 2713 2716 2717 2725 2730 2732 2739 2741}
        hdisk154 {2742 2743 2744 2747 2748 2749 2750 2752 2755 2757 2758}
        hdisk155 {2759 2761 2762 2763 2765 2766 2768 2769 2771 2772 2774}
        hdisk220 {2687 2688 2689 2693 2694 2695 2697 2699 2700 2701 2703}
        hdisk156 {2690 2691 2692 2696 2698 2702 2704 2705 2706 2707 2708}
        hdisk157 {2712 2714 2715 2718 2719 2720 2721 2722 2723 2724 2726}
        hdisk158 {2727 2728 2729 2731 2733 2734 2735 2736 2737 2738 2740}
        hdisk159 {2745 2746 2751 2753 2754 2756 2760 2764 2767 2770 2773}

        D1F28V1ITEM  D1F28V1WARE  D1F28V1DIST  D1F28V1NORI  D1F28V1CSTI28
        D1F28V1ORL28  D1F28V1STK28  D1F28V1CST28  D1F28V1ORDI  D1F28V1ORD
        D1F28V1NOR  D1F28V1HIST  D1F28V2ITEM  D1F28V2WARE  D1F28V2DIST
        D1F28V2NORI  D1F28V2CSTI28  D1F28V2ORL28  D1F28V2STK28  D1F28V2CST28
        D1F28V2ORDI  D1F28V2ORD  D1F28V2NOR  D1F28V2HIST
```

Here's the disk mapping for log:

The last pdisk of each hdisk is the hot spare.

```
Adaptor  Hdisk Pdisk

ssa14

      hdisk252 {1361 1373 1367 1360 1350 1370 1348 1368)
      hdisk253 {1355 1385 1372 1369 1383 1389 1386 1391}
      hdisk254 {1365 1357 1345 1387 1375 1362 1354 1364}
      hdisk255 {1352 1374 1358 1379 1378 1356 1390 1376}
      hdisk256 {1353 1351 1359 1381 1377 1349 1363 1371}
      hdisk257 {1382 1384 1344 1380 1366 1346 1388 1347}

ssa20

      hdisk228 {1878 1877 1876 1874 1913 1912 1875 1894}
      hdisk229 {1900 1899 1873 1879 1901 1906 1888 1890}
      hdisk230 {1895 1872 1893 1882 1885 1904 1883 1886}
      hdisk249 {1909 1908 1892 1917 1916 1915 1919 1918}
      hdisk250 {1880 1881 1884 1889 1897 1905 1911 1898}
      hdisk251 {1887 1891 1896 1902 1903 1910 1914 1907}

      LV DB2LOGLV stripped across 12 hdisks.
```

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 DB2 UDB. DB2 UDB is a relational DBMS. DB2 remote stored procedures and embedded SQL statements were used. The DB2 stored procedures were invoked via SQL CALL statements. Both the client application and stored procedures were written in embedded C code.

5.4 Partitions/Replications Mapping

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

The Customer, Order-Line, and Stock tables were horizontally partitioned using the warehouse ID. The specifics of the distribution of partitioned and non-partitioned tables across the physical media can be found in Table 5-2.

5.5 60 day space calculations

IBM eServer pSeries 690 Turbo Model 7040-681

60 Days Space Computation

All Data sizes are in MB unless otherwise stated

Warehouses	53,200	**tpmC**	680,613.12	**tpmC/W**	12.79

Table	**Rows**	**Data**	**Index**	**5% Space**	**Total Space**
History	1,596,000,000	98,208	0	0	98,208
Orders	1,596,000,000	58,546	44,548	0	103,094
Order-line	15,960,647,808	1,047,974	0	0	1,047,974
New-order	53,200	9,314	11,219	0	20,533
Warehouse	532,000	21	0	1	22
District	100,000	61	0	3	64
Item	478,800,000	11	0	1	11
Customer	1,596,000,000	1,247,008	77,000	66,200	1,390,208
Stock	5,320,000,000	1,731,884	0	86,594	1,818,478

Free Space	488,645
Dynamic Space	1,214,042
Static Space	3,208,783
Daily Growth	248,509
Daily Spread	115,881
60 Days MB	25,072,187
60 Days GB	24,485

Log File Storage Requirement:		
Log Pages Written (KB)	52,545,872.00	log file storage used in a steady state 30 minute interval
Total N-O Txn	20,300,331.90	New Order transactions completed during the same 30 minute interval
Log per N-O txn	2.59	KB of log storage used per New Order transaction
8 Hour Log (GB)	806.45	8-hour log storage required

Disks Type	**Disk Formatted Capacity (GB)**	**SUT # of Disks**	**SUT Capacity (GB)**	**Priced # of Disks**	**Priced Capacity (GB)**
DB SSA 18.2 GB	16.969	2,464	41,811	2,464	41,811
LOG SSA RAID5 (7x18.2GB)	101.813	12	1,222	12	1,222
OS SCSI 18.2 GB					33.94
Priced Space					43,067

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 lists 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 lists 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 %	0.12	0.08	0.13	0.05/0.23	0.08	0
Average	0.08	0.05	0.09	0.03/0.28	0.05	0
Maximum	17.99	24.02	12.45	14.26/11.76	11.92	15.11
			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	120.21	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	18.12	3.08	2.08	2.08	2.08	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

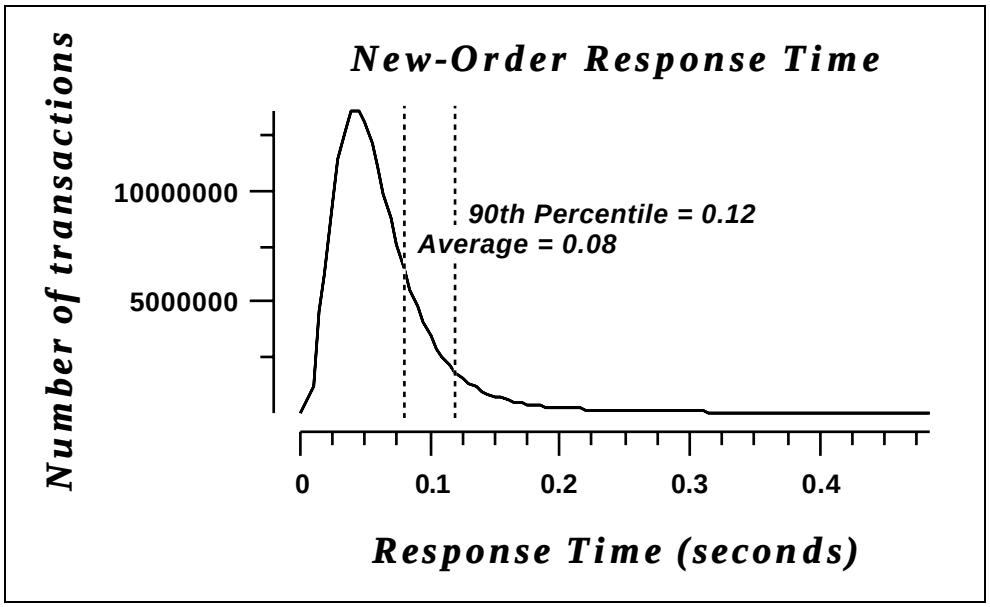


Figure 6-3-2. IBM eServer pseries 690 Turbo Model 7040-681 Payment Response Time Distribution

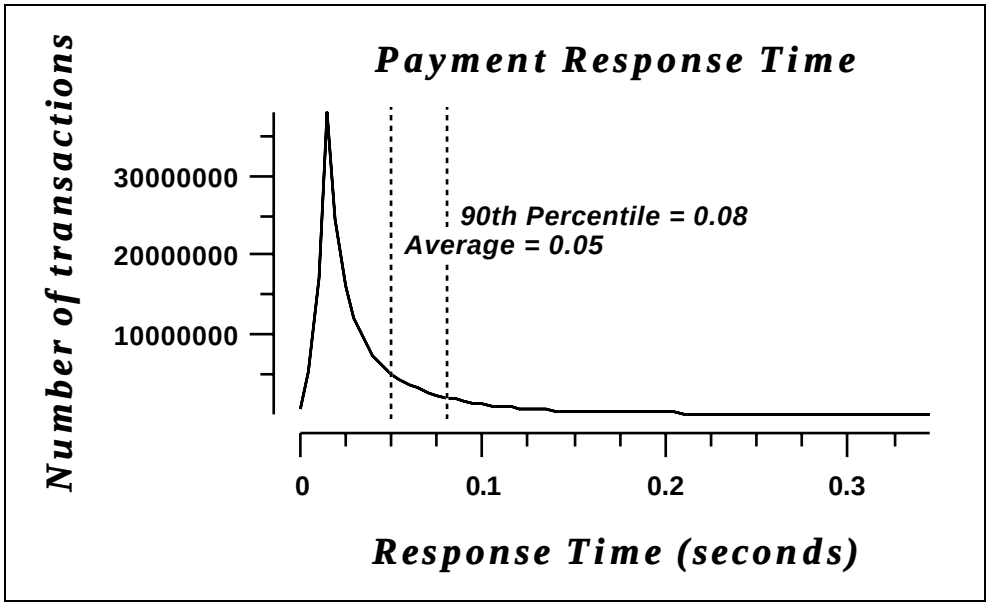


Figure 6-3-3. IBM eServer pseries 690 Turbo Model 7040-681 Order-Status Response Time Distribution

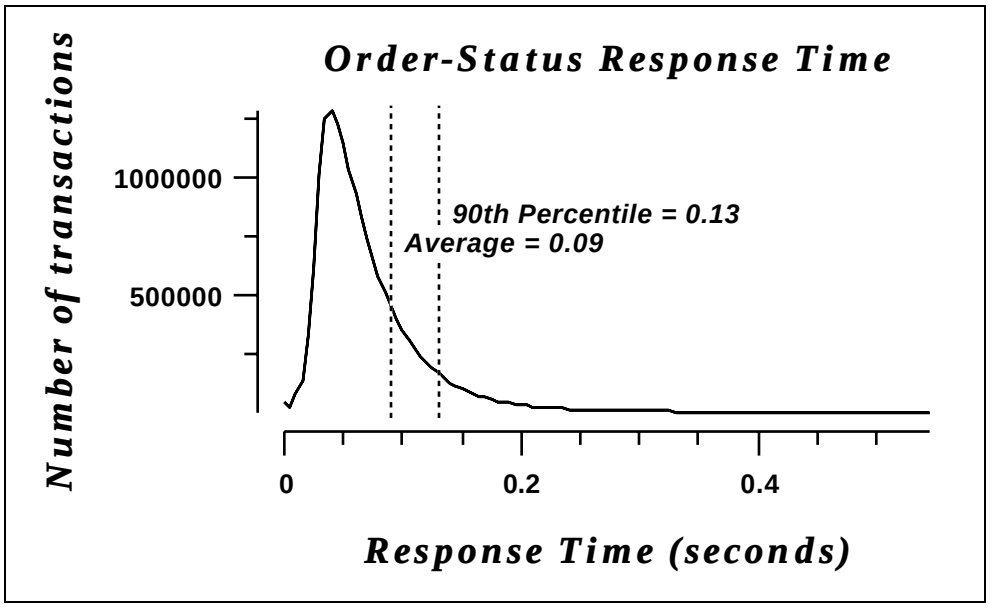


Figure 6-3-4. IBM eServer pSeries 690 Turbo Model 7040-681 Delivery (Interactive) Response Time Distribution

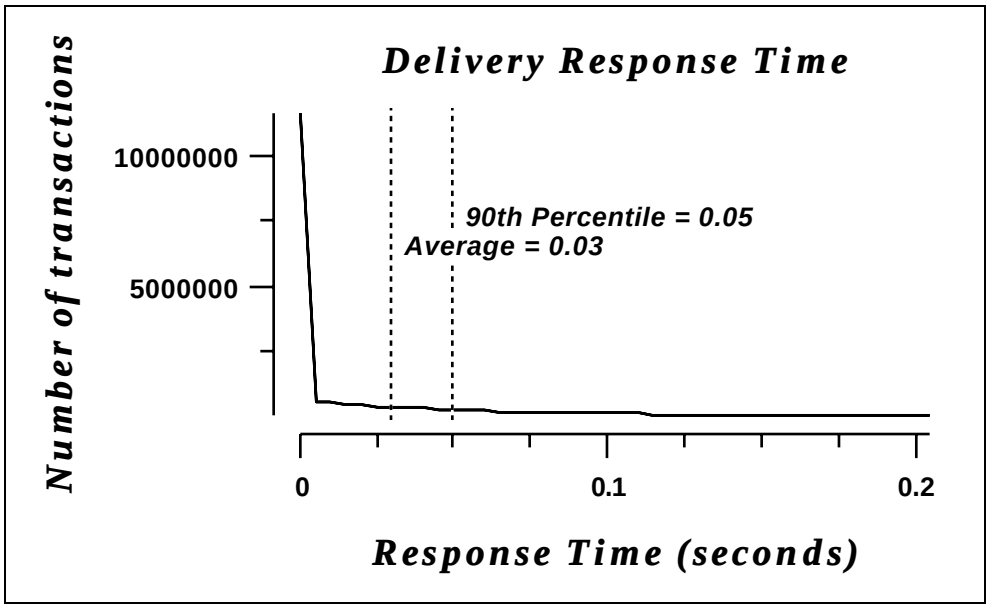


Figure 6-3-5. IBM eServer pSeries 690 Turbo Model 7040-681 Delivery (Deferred) Response Time Distribution

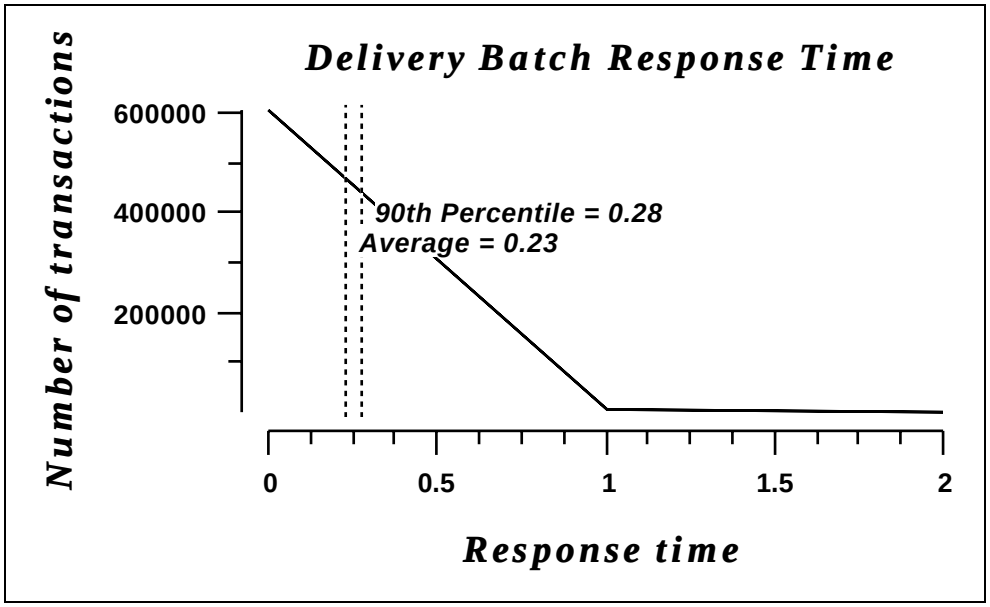
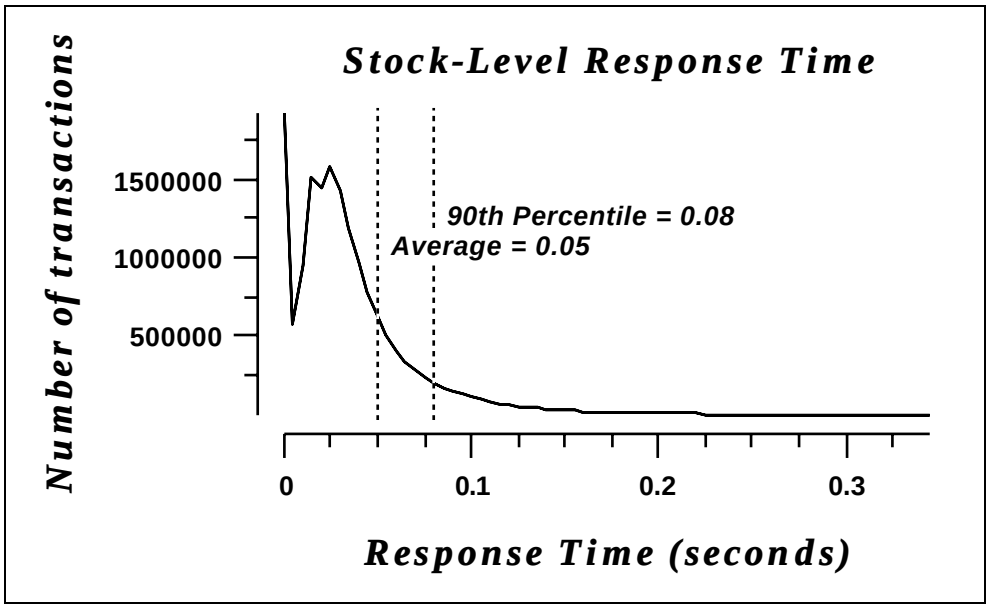


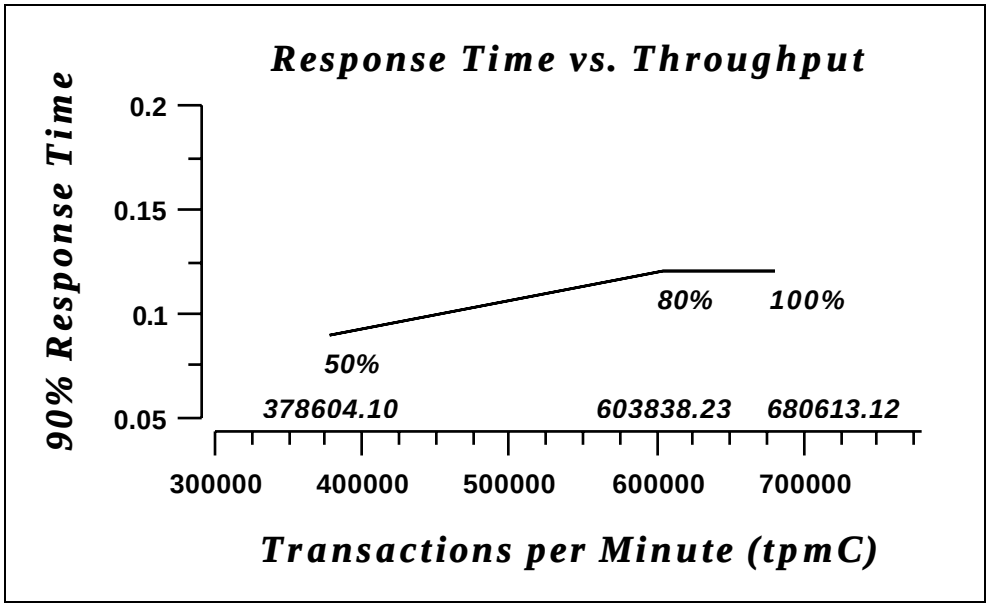
Figure 6-3-6. IBM eServer pSeries 690 Turbo Model 7040-681 Stock Level Response Time Distribution



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.

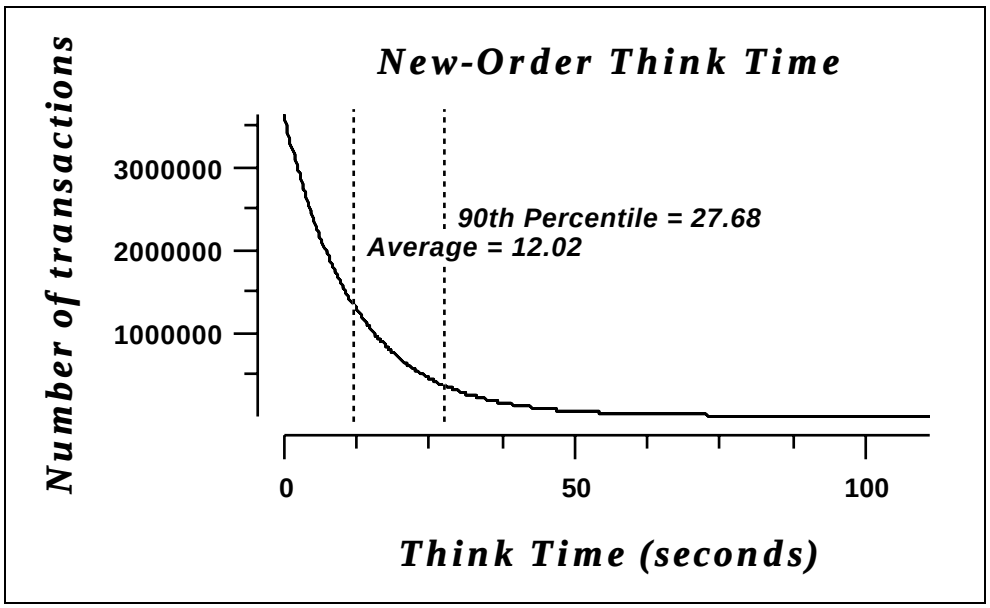
Figure 6-4-1. IBM eServer pSeries 690 Turbo Model 7040-681 New-Order Response Time vs. Throughput



6.5 Think Time Frequency Distribution

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

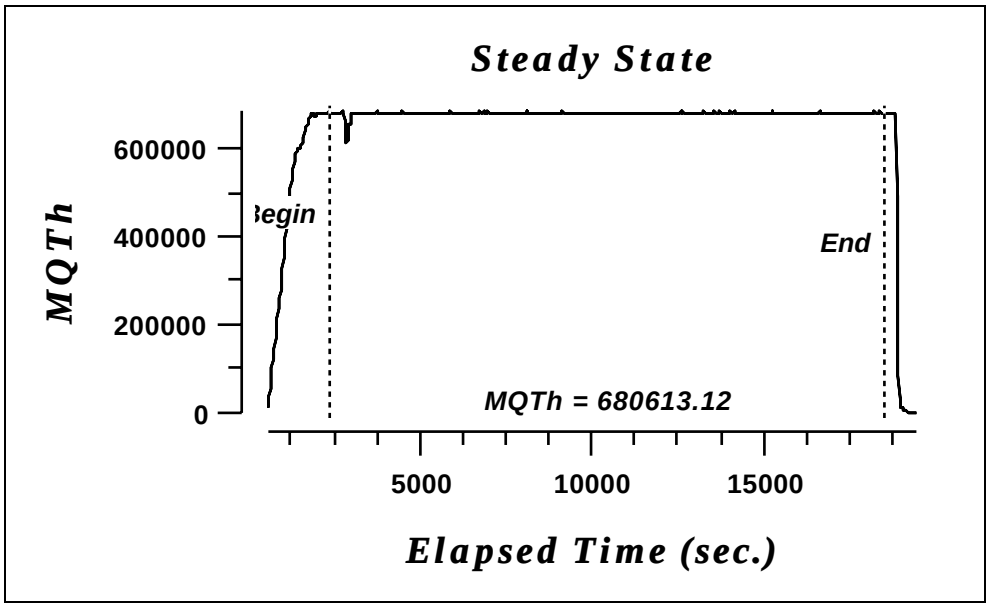
Figure 6-5-1. IBM eServer pSeries 690 Turbo Model 7040-681 New-Order Think Time Distribution



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 user ramp-up phase is clearly visible on the graph above. 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 maintained a steady state during 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.

A 4 hours 30 minutes measurement interval was used to guaranty that all work normally performed during an 8-hours sustained test are included in the reported throughput.

6.8.1 Transaction Flow

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

Tuxedo was used as a transaction manager (TM). Each transaction was divided into two programs. The front end program handled all screen I/O while the back end program handled all database operations. Both the front end and back end programs ran on the client system. The front end program communicates with the back end program through Tuxedo messages. The back end program communicates with the Server system over Ethernet using DB2 embedded SQL calls. Besides calling Tuxedo functions for user connection and message communication, all other functions are transparent to the application code. Tuxedo routes the transaction and balances the load according to the options defined in the Tuxedo configuration file listed in Appendix B.2. The transaction flow is described below.

- When Tuxedo boots up, it creates one or more server process(es) for each transaction. Several server processes were defined in the Tuxedo configuration file.
- Each TPC-C user invokes the TPC-C client (front end) program.

- The TPC-C client program 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 that transaction.
- The TPC-C client program accepts and checks all values entered by the user and transmits those values to one of the TPC-C back end programs. The transmission is performed through a Tuxedo function call. Each TPC-C back end program has a "service-name". This service-name is specified whenever the TPC-C client program requests a Tuxedo service. Tuxedo routes that message according to the service-name and the information defined in the Tuxedo configuration file.
- A TPC-C back end server program receives a message from its queue and proceeds to execute all database operations related to the service-name specified. All the transaction information entered on the user terminal is contained in the Tuxedo message.
- Once the transaction is committed, the TPC-C back end server program loads the message buffer with the transaction output and returns control to Tuxedo.
- Tuxedo manager routes the message back to the TPC-C client program..

6.8.2 Database Transaction

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

Using embedded SQL calls, the TPC-C back-end program interacts with DB2 UDB 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.

DB2 UDB Server proceeds to update the database as follows:

When DB2 UDB 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, DB2 UDB Server will make space by flushing some modified pages to disk. Modified pages are also written to disk as part of the "Soft" checkpoint to ensure that no updates remain unflushed for longer than the allowed time. 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

DB2 UDB uses a write-ahead-logging protocol to guarantee recovery. This protocol uses "Soft" checkpoint to write least-recently-used database pages to disk independent of transaction commit. However, enough log information to redo/undo the change to a database pages is committed to disk before the database page itself is written. This protocol therefore renders checkpoint unnecessary for DB2 UDB. For a more detailed description of the general principles of the write-ahead-logging protocol, see the IBM research paper, "ARIES: A Transaction Recovery Method Supporting Fine Granularity Locking and Partial Rollbacks Using Write-Ahead Logging," by C. Mohan, Database Technology Institute, IBM Almaden Research Center.
(http:// portal.acm.org/citation.cfm?id=128770&coll=portal&dl=ACM&CFID=10343790&CFTOKEN=42047146)

6.9 Measurement Interval

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

A 4 hours 30 minutes measurement interval was used. 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 63 eServer pSeries 6F1 emulates a network of 532,000 eServer pSeries 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 704 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 100 Megabits per second Full Duplex.. Each LAN segment in the IBM eServer pSeries 690 Turbo Model 7040-681 configuration connected 704 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.

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 sheet for this disclosure is contained in the executive summary pages. 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 52% from the list price, based on the dollar value of this configuration only.

For assistance with any of these prices or their applicability to any customer's requirements, please contact the following individuals:

Hardware:

William R. Casey,
eServer pSeries Offering Manager
wrcasey@us.ibm.com
(512) 838-1422

DB2:

Paul Rivot,
Director, Database Servers and Business Intelligence Software
privot@us.ibm.com
(914) 766-1325

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
IBM eServer pSeries 690 Turbo Model 7040-681	May 30, 2003
AIX APAR IY43549 for AIX V5.2	November 8, 2003
DB2 UDB 8.1.4	November 8, 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	680,613.12	$7,574,961	$11.13	All HS/SW available as shown in Section 8.3

Please refer to the price list on the Executive Summary page for details.

. 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

Benchmark Sponsor:	William A. Maron IBM eServer Performance 11501 Burnet Road Austin, Texas 78758	Berni Schiefer IBM DB2 Performance 8200 Warden Ave Markham, Ontario, L6G 1C7

May 8, 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:	**IBM UDB DB2 8.1**
Transaction Manager:	**BEA Tuxedo 8.0**

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: IBM eServer pSeries 690 Turbo Model 7040-681				
32 x POWER4+ (1.7 GHz)	512 GB Main (128MB L3 Cache per MCM)	2563 x 18.2 GB	0.12 Seconds	680,613.12
Twenty Seven (27) Clients: eServer pSeries Model 6E4 (Specification for each)				
4 x POWER4 (1.0 GHz)	16 GB Main (32 MB L3 Cache per CPU)	1 x 18.2 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC's 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 4 hours 30 minutes
- Continuous checkpoints were active 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

Appendix A: TPC-C Application Source

A.1 Client/Terminal Handler code

Makefile

```
#####################################################################
## Project                                  : DB/2 TPCC
## Component                : TUX Client
## Title                                                : db2Client
## Copyright     : 2003 IBM Corporation
## Classification : All rights reserved
##
## History       :
##              Developed at IBM Austin by the AIX RS/6000
##              performance group.
##
## Comments      :
##
#####################################################################

#####################################################################
## Declare the module name & path & Local Vars
##                          NB the metafile name must not exceed eight characters and
##                                          is unprefiexed
#####################################################################

PROGRAM                 = tuxClient
MODULE                  = /home/tpcc/toraus/db2kit/db2client
COMMON                  = /home/tpcc/toraus/db2kit/db2common
EXEHOME                 = /home/tpcc/toraus/db2kit/db2inf
TUXINC      = /usr/local/bea/tuxedo8.0/include
INSTALLDIR= $(EXEHOME)
MAKEFILE                = makefile

SETDEBUG    = -g
SETOPT      = -O3 -qinlglue -qproto -qunroll=8

#####################################################################
## Application options
#####################################################################

LOCALSESSION            = -DLOCAL_SESSION_DATA

#####################################################################
## Compiler settings
#####################################################################

CC                                                  = xlC_r
CCC                                 = xlC_r
INC                                 = -I/home/tpcc/toraus/db2kit/db2client \
                                                    -I/home/tpcc/toraus/db2kit/db2common \
                                                    -I/usr/local/bea/tuxedo8.0/include
CFLAGS                  = -I/home/tpcc/db2kit/db2client \
                                                    -I/home/tpcc/db2kit/db2common \
                                                    -I/usr/local/bea/tuxedo8.0/include \
                                                    -D__UNIX__ -D__AIXPOWER__
-D__DB2__ \
                                                    -+ -qnotempinc -qcpluscmt -qchars=unsigned \
                                                    -DLOCAL_SESSION_DATA
CCFLAGS                 = -I/home/tpcc/toraus/db2kit/db2client \
                                                    -I/home/tpcc/toraus/db2kit/db2common \
                                                    -I/usr/local/bea/tuxedo8.0/include \
                                                    -D__UNIX__ -D__AIXPOWER__
-D__DB2__ \
                                                    -+ -qnotempinc -qcpluscmt -qchars=unsigned \
                                                    -DLOCAL_SESSION_DATA
LPATH                               =
LIBS                                = -lc -lm -lsvld -ls -libmuis
LFLAGS                  = -lc -lm -lsvld -ls -libmuis -bh:5

#####################################################################
## define what componects are required for module
#####################################################################

#####################################################################
## Module Source files (.cpp)
##                          NB the source files must be listed in the same order as
##                                          the object files
#####################################################################

PRGSRCS                 = \
                                                                    $(MODULE)/tuxClientInf.cpp \
                                                                    $(MODULE)/tuxClient.cpp \

$(MODULE)/tuxClientInfUtilities.cpp \

$(MODULE)/tuxClientCommon.cpp \

$(MODULE)/tuxClientMonitor.cpp \

$(MODULE)/tuxClientListen.cpp \

$(MODULE)/tuxClientScreen.cpp \

$(MODULE)/tuxClientTerminal.cpp \
                                                                                                $(MODULE)/tuxClientField.cpp
\

$(MODULE)/tuxClientThread.cpp \

$(MODULE)/tuxClientNullDB.cpp

#####################################################################
## Module Object files (.o)
##                          NB the object files must be listed in the same order as
##                                          the source files
#####################################################################

PRGOBJS                 = tuxClientInf.o \
                                                                    tuxClient.o \
                                                                    tuxClientInfUtilities.o \
                                                                    tuxClientCommon.o \
                                                                    tuxClientMonitor.o \
                                                                    tuxClientListen.o \
                                                                    tuxClientScreen.o \
                                                                    tuxClientTerminal.o \
                                                                    tuxClientField.o \
                                                                    tuxClientThread.o \
                                                                    tuxClientNullDB.o

#####################################################################
## standard makefile rules
#####################################################################

#####################################################################
## Setup lists of files for processing
#####################################################################

SOURCECODE              = $(PRGSRCS)

OBJECTCODE              = $(PRGOBJS)

#####################################################################
## File Sufixes
##                          .sxx  - sqc metafile
##                          .sqc            - imbeded sql
##                          .cpp            - c++ source code
##                          .c                              - c source code
##                          .o                              - object code
#####################################################################

.SUFFIXES: .cpp .c .o

#####################################################################
## Build Targets
##                          all                             - build module
##                          link                            - link module
##                          strip                           - strip module
##                          install         - install module
##                          clean                           - clean module working directory
##                          depends         - update makefile dependencies
#####################################################################

all: $(PROGRAM)

$(PROGRAM): $(OBJECTCODE)
                ar -v -r $(PROGRAM).a $(PRGOBJS)
                buildclient -o $(PROGRAM) -f $(PROGRAM).a

link:
                ar -v -r $(PROGRAM).a $(PRGOBJS)
                buildclient -o $(PROGRAM) -f $(PROGRAM).a

strip:
                @-strip $(PROGRAM)
                @-strip $(PROGRAM).a

install:
                @-rm -f $(INSTALLDIR)/$(PROGRAM)
                @-rm -f $(INSTALLDIR)/$(PROGRAM).a
                @-cp -f $(PROGRAM) $(INSTALLDIR)
                @-cp -f $(PROGRAM).a $(INSTALLDIR)
                @-chmod 755 $(INSTALLDIR)/$(PROGRAM)
                @-chmod 755 $(INSTALLDIR)/$(PROGRAM).a

clean:
                @-rm -f $(OBJECTCODE)
                @-rm -f $(PROGRAM)
                @-rm -f $(PROGRAM).a

depends:
                @-/home/tpcc/toraus/db2kit/db2inf/depend -m$(MAKEFILE) $(INC) $(SOURCECODE)

#####################################################################
##              Source to Object (.cpp -> .o)
```

```
#####################################################################

.cpp.o:
	$(CCC) -q32 -c $< -o $@ $(INC) $(CCFLAGS)

#####################################################################
##		Source to Object (.c -> .o)
#####################################################################

.c.o:
	$(CCC) -q32 -c $< -o $@ $(INC) $(CCFLAGS)

#####################################################################
##		Makefile End
#####################################################################

# DO NOT DELETE THIS LINE -- dependency generator depends on it. Every after this line is replaced

tuxClientInf.o: \
	/home/tpcc/toraus/db2kit/db2client/tuxClientInf.cpp \
	./tuxClientInf.hpp \
	/home/tpcc/toraus/db2kit/db2common/commonTuxDB2.hpp \
	/usr/local/bea/tuxedo8.0/include/atmi.h \
	/usr/local/bea/tuxedo8.0/include/tmenv.h \
	/home/tpcc/toraus/db2kit/db2common/db2tpcc.h \
	/home/tpcc/toraus/db2kit/db2common/lval.h \
	./tuxClientNullDB.hpp

tuxClient.o: \
	/home/tpcc/toraus/db2kit/db2client/tuxClient.cpp \
	./tuxClient.hpp \
	/home/tpcc/toraus/db2kit/db2common/commonTuxDB2.hpp \
	/usr/local/bea/tuxedo8.0/include/atmi.h \
	/usr/local/bea/tuxedo8.0/include/tmenv.h \
	/home/tpcc/toraus/db2kit/db2common/db2tpcc.h \
	/home/tpcc/toraus/db2kit/db2common/lval.h \
	./tuxClientCommon.hpp \
	./tuxClientInf.hpp \
	./tuxClientNullDB.hpp \
	./tuxClientInfUtilities.hpp \
	./tuxClientMonitor.hpp \
	./tuxClientThread.hpp \
	./tuxClientListen.hpp \
	./tuxClientScreen.hpp \
	./tuxClientTerminal.hpp \
	./tuxClientTerminal.inl \
	./tuxClientField.hpp \
	./tuxClientScreen.inl

tuxClientInfUtilities.o: \
	/home/tpcc/toraus/db2kit/db2client/tuxClientInfUtilities.cpp \
	./tuxClientInfUtilities.hpp \
	/home/tpcc/toraus/db2kit/db2common/commonTuxDB2.hpp \
	/usr/local/bea/tuxedo8.0/include/atmi.h \
	/usr/local/bea/tuxedo8.0/include/tmenv.h \
	/home/tpcc/toraus/db2kit/db2common/db2tpcc.h \
	/home/tpcc/toraus/db2kit/db2common/lval.h \
	./tuxClientInf.hpp \
	./tuxClientNullDB.hpp \
	./tuxClientMonitor.hpp \
	./tuxClientCommon.hpp \
	./tuxClientThread.hpp

tuxClientCommon.o: \
	/home/tpcc/toraus/db2kit/db2client/tuxClientCommon.cpp \
	./tuxClientCommon.hpp \
	/home/tpcc/toraus/db2kit/db2common/commonTuxDB2.hpp \
	/usr/local/bea/tuxedo8.0/include/atmi.h \
	/usr/local/bea/tuxedo8.0/include/tmenv.h \
	/home/tpcc/toraus/db2kit/db2common/db2tpcc.h \
	/home/tpcc/toraus/db2kit/db2common/lval.h \
	./tuxClientInf.hpp \
	./tuxClientNullDB.hpp

tuxClientMonitor.o: \
	/home/tpcc/toraus/db2kit/db2client/tuxClientMonitor.cpp \
	./tuxClientMonitor.hpp \
	/home/tpcc/toraus/db2kit/db2common/commonTuxDB2.hpp \
	/usr/local/bea/tuxedo8.0/include/atmi.h \
	/usr/local/bea/tuxedo8.0/include/tmenv.h \
	/home/tpcc/toraus/db2kit/db2common/db2tpcc.h \
	/home/tpcc/toraus/db2kit/db2common/lval.h \
	./tuxClientCommon.hpp \
	./tuxClientInf.hpp \
	./tuxClientNullDB.hpp \
	./tuxClientInfUtilities.hpp \
	./tuxClientThread.hpp

tuxClientListen.o: \
	/home/tpcc/toraus/db2kit/db2client/tuxClientListen.cpp \
	./tuxClientListen.hpp \
	/home/tpcc/toraus/db2kit/db2common/commonTuxDB2.hpp \
	/usr/local/bea/tuxedo8.0/include/atmi.h \
	/usr/local/bea/tuxedo8.0/include/tmenv.h \
	/home/tpcc/toraus/db2kit/db2common/db2tpcc.h \
	/home/tpcc/toraus/db2kit/db2common/lval.h \
	./tuxClientCommon.hpp \
	./tuxClientInf.hpp \
	./tuxClientNullDB.hpp \
	./tuxClientInfUtilities.hpp \
	./tuxClientMonitor.hpp \
	./tuxClientThread.hpp \
	./tuxClientScreen.hpp \
	./tuxClientTerminal.hpp \
	./tuxClientTerminal.inl \
	./tuxClientField.hpp \
	./tuxClientScreen.inl

tuxClientScreen.o: \
	/home/tpcc/toraus/db2kit/db2client/tuxClientScreen.cpp \
	./tuxClientScreen.hpp \
	/home/tpcc/toraus/db2kit/db2common/commonTuxDB2.hpp \
	/usr/local/bea/tuxedo8.0/include/atmi.h \
	/usr/local/bea/tuxedo8.0/include/tmenv.h \
	/home/tpcc/toraus/db2kit/db2common/db2tpcc.h \
	/home/tpcc/toraus/db2kit/db2common/lval.h \
	./tuxClientCommon.hpp \
	./tuxClientInf.hpp \
	./tuxClientNullDB.hpp \
	./tuxClientTerminal.hpp \
	./tuxClientMonitor.hpp \
	./tuxClientInfUtilities.hpp \
	./tuxClientThread.hpp \
	./tuxClientTerminal.inl \
	./tuxClientField.hpp \
	./tuxClientScreen.inl

tuxClientTerminal.o: \
	/home/tpcc/toraus/db2kit/db2client/tuxClientTerminal.cpp \
	./tuxClientTerminal.hpp \
	/home/tpcc/toraus/db2kit/db2common/commonTuxDB2.hpp \
	/usr/local/bea/tuxedo8.0/include/atmi.h \
	/usr/local/bea/tuxedo8.0/include/tmenv.h \
	/home/tpcc/toraus/db2kit/db2common/db2tpcc.h \
	/home/tpcc/toraus/db2kit/db2common/lval.h \
	./tuxClientCommon.hpp \
	./tuxClientInf.hpp \
	./tuxClientNullDB.hpp \
	./tuxClientMonitor.hpp \
	./tuxClientInfUtilities.hpp \
	./tuxClientThread.hpp \
	./tuxClientTerminal.inl

tuxClientField.o: \
	/home/tpcc/toraus/db2kit/db2client/tuxClientField.cpp \
	./tuxClientField.hpp \
	/home/tpcc/toraus/db2kit/db2common/commonTuxDB2.hpp \
	/usr/local/bea/tuxedo8.0/include/atmi.h \
	/usr/local/bea/tuxedo8.0/include/tmenv.h \
	/home/tpcc/toraus/db2kit/db2common/db2tpcc.h \
	/home/tpcc/toraus/db2kit/db2common/lval.h \
	./tuxClientCommon.hpp \
	./tuxClientInf.hpp \
	./tuxClientNullDB.hpp \
	./tuxClientTerminal.hpp \
	./tuxClientMonitor.hpp \
	./tuxClientInfUtilities.hpp \
	./tuxClientThread.hpp \
	./tuxClientTerminal.inl

tuxClientThread.o: \
	/home/tpcc/toraus/db2kit/db2client/tuxClientThread.cpp \
	./tuxClientThread.hpp \
	/home/tpcc/toraus/db2kit/db2common/commonTuxDB2.hpp \
	/usr/local/bea/tuxedo8.0/include/atmi.h \
	/usr/local/bea/tuxedo8.0/include/tmenv.h \
	/home/tpcc/toraus/db2kit/db2common/db2tpcc.h \
	/home/tpcc/toraus/db2kit/db2common/lval.h \
	./tuxClientCommon.hpp \
	./tuxClientInf.hpp \
	./tuxClientNullDB.hpp

tuxClientNullDB.o: \
	/home/tpcc/toraus/db2kit/db2client/tuxClientNullDB.cpp \
	./tuxClientNullDB.hpp \
	/home/tpcc/toraus/db2kit/db2common/commonTuxDB2.hpp \
	/usr/local/bea/tuxedo8.0/include/atmi.h \
	/usr/local/bea/tuxedo8.0/include/tmenv.h \
	/home/tpcc/toraus/db2kit/db2common/db2tpcc.h \
	/home/tpcc/toraus/db2kit/db2common/lval.h
```

tuxClient.cpp

```
/*
*********************************************************************
** Project                      : AIX DB/2 TPC-C
** Component : TPC-C/Client
** Name                         : tuxClient.cpp
** Title                        : Abstraction layer for TUX client
*********************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification               : IBM Internal Use Only
**
** History                      :
**          Developed at IBM Austin by the AIX RS/6000
```

```
**              performance group.
**
** Comments                                    :
**                                                                    Please read
below
**
** Notes                                                  :
** This is the main client program for the TPCC benchmark using Tuxedo.
**
** The client program is multi-threaded: there is one thread for each
** terminal and one thread to process incoming connections.
**
**
*****************************************************************
*/

#include <tuxClient.hpp>

/*
*****************************************************************
** Name       : main
** Description :
**            Start tux client
** Parameters  :
**            int - argument count
**                                                                char ** - arguments
** Returns     :
**            int - return code
** Comments    :
**
** Notes      :
**
*****************************************************************
*/

extern "C" int main(int argc,char **argv)
{
                char            logmsg[MAX_MSG_LEN];

                // Process command line
                check_parms(argc,argv);
                sprintf(logmsg,"Client %d starting.\n",user_id);
                err_printf(logmsg);

                // Setup user
                init_tux_client(user_id);
                enroll_client(user_id);

                // Open logfile
                #if DEBUG_LOG
                                logtpcc = fopen(log_file_name,"w");
                #endif

                #if ERROR_LOG
                                errtpcc = fopen(err_file_name,"w");
                #else
                                errtpcc = stderr;
                #endif

                #if COMM_LOG
                                comlog  = fopen(com_file_name,"w");
                #endif

                print_header(argc,argv);

                // listner which will not return
                make_connections((int) user_port);

                // exit
                exit_program(0); return OK;
}

/*
*****************************************************************
** Name       : check_params
** Description :
**            Retrive and validate parameters
** Parameters  :
**            int - argument count
**                                                                char ** - arguments
** Returns     :
**            None
** Comments    :
**                                                                User must supply user_id as a
parm and all other
**                                                                parameters as environment
variables.
**
** Notes      :
**                                                                Not all the parameters are
relevant for this executable.
**
*****************************************************************
*/

void check_parms(int argc,char **argv)
{
                int             next_arg                = 1;
                int             errors                  = 0;
                int             print_help      = 0;

                char            *progName;
                char            *home_dir       = getenv("TPCC_HOME");
                char            *hostname       = getenv("TPCC_HOST");

                // Defaults
                user_id = -1; result_dir = ".";
                strcpy(host_name,hostname);

                // process parameter list
                while (next_arg < argc)
                {
                                if(!strcasecmp("-debug",argv[next_arg]))
                                {
                                                debug = 1;
                                                next_arg++; continue;
                                }

                                if(!strcasecmp("-dir",argv[next_arg]))
                                {
                                                result_dir = argv[++next_arg];
                                                next_arg++; continue;
                                }

                                if(!strcasecmp("-log",argv[next_arg]))
                                {
                                                logtrans = 1;
                                                next_arg++; continue;
                                }

                                if(!strcasecmp("-id",argv[next_arg]))
                                {
                                                user_id = atol(argv[++next_arg]);
                                                next_arg++; continue;
                                }

                                if(!strcasecmp("-host",argv[next_arg]))
                                {
                                                strcpy(host_name,argv[++next_arg]);
                                                next_arg++; continue;
                                }

                                if(!strcasecmp("-port",argv[next_arg]))
                                {
                                                user_port = atol(argv[++next_arg]);

                                                if (user_id < 0) user_id = user_port;
                                                next_arg++; continue;
                                }

                                if(!strcasecmp("-security",argv[next_arg]))
                                {
                                                useSecurity = TRUE;
                                                next_arg++; continue;
                                }

                                if(!strcasecmp("-noSecurity",argv[next_arg]))
                                {
                                                useSecurity = FALSE;
                                                next_arg++; continue;
                                }

                                if(!strcasecmp("-null",argv[next_arg]))
                                {
                                                null_db = 1;
                                                next_arg++; continue;
                                }

                                if(!strcasecmp("-lock",argv[next_arg]))
                                {
                                                client_lock_handles = atol(argv[++next_arg]);
                                }

                                if(!strcasecmp("-help",argv[next_arg]))
                                {
                                                print_help = 1; break;
                                }

                                printf("invalid parameter: %s\n",argv[next_arg]);
                                print_help = 1; break;
                }

                if(!print_help)
                {
                                // validate bind address
                                if(!host_name)
                                {
                                                printf("   Must specif a host name or set environemt
TPCC_HOST\n");
                                                print_help = 1;
                                }

                                // validate user id
                                if (user_id < 0)
                                {
                                                printf("  Missing User Id\n");
                                                print_help = 1;
                                }
```

```
        }

        // Provide some help to the user
        if (print_help)
        {
                progName = strrchr(argv[0], '/');
                progName = (progName ? progName + 1 : argv[0]);

                printf("\nusage:\n You can specify the following in any order\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");
                printf("   -port          port number to listen on\n");
                printf("   -host          hostname to use\n");
                printf("   -help          print this help text\n");

                exit(-1);
        }

        // log update
        sprintf(log_file_name, "%s/%s/C.%s.%d_log",
        home_dir ? home_dir : "/home/tpcc",
        LOG_FILE_DIR,
        host_name ? host_name : "host", user_id);

        sprintf(err_file_name, "%s/%s/C.%s.%d_err",
        home_dir ? home_dir : "/home/tpcc",
        LOG_FILE_DIR,
        host_name ? host_name : "host", user_id);

        sprintf(com_file_name, "%s/%s/C.%s.%d_com",
        home_dir ? home_dir : "/home/tpcc",
        LOG_FILE_DIR,
        host_name ? host_name : "host", user_id);
        return;
}

/*
*****************************************************************
** Name       : print_header
** Description :
**          Print log header record
** Parameters  :
**          int - argument count
**                                                        char ** - arguments
** Returns     :
**          None
** Comments    :
**
** Notes       :
**
*****************************************************************
*/

void print_header(int argc, char *argv[])
{
        char            logmsg[MAX_MSG_LEN];

        if (!logtpcc)
                return;

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

        sprintf(logmsg,"Params: ");
        logprintf(logmsg);

        // print each parameter
        for(int i=0;i<argc;i++)
                fprintf(logtpcc,"%s ",argv[i]);

        fprintf(logtpcc,"\n");
        fflush(logtpcc);

        return;
}

/////////////////////////////////////////////////////////////////////
```

tuxClient.hpp

```
/*
*****************************************************************
** Project                  : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                                  : tuxClient.hpp
** Title                                 : Tux Client start
*****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History       :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                              :
**
*****************************************************************
*/

#ifndef __tuxClient_hpp__
#define __tuxClient_hpp__

#include <commonTuxDB2.hpp>
#include <tuxClientCommon.hpp>

#include <tuxClientInf.hpp>
#include <tuxClientInfUtilities.hpp>
#include <tuxClientListen.hpp>
#include <tuxClientMonitor.hpp>

/////////////////////////////////////////////////////////////////////
// Function Definitions
/////////////////////////////////////////////////////////////////////

extern "C" int main(int argc, char **argv);

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

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

#endif
```

tuxClientCommon.cpp

```
/*
*****************************************************************
** Project                  : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                                  : tuxClientCommon.cpp
** Title                                 : Tux client common helpers
*****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History       :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                              :
**
*****************************************************************
*/

#include <tuxClientCommon.hpp>

/////////////////////////////////////////////////////////////////////
// Externals
/////////////////////////////////////////////////////////////////////

int                                                                     user_id
                                        = 0;
int                                                                     useSecurity
                        = FALSE;
int                                                                     debug
                                                = 0;
int                                                                     info_list_len
                = 0;
int                                                                     warehouse_offset
                = 0;
int                                                                     logtrans
                                        = 0;
int
client_lock_handles     = 0;
int                                                                     user_port
                        = 4011;
int                                                                     consecutiveErrors
                = 0;
int                                                                     iStatsFrequency
                = 1;
int                                                                     envRetrieval
                = 0;
int
num_active_threads      = 0;

char                                                                    host_name[256];

char
log_file_name[256];
char
err_file_name[256];
char
com_file_name[256];
```

```
char                                                    *user_code
                        = "C";
char                                                    *result_dir
                        = NULL;
char                                                    *LOG_FILE_DIR
                        = "runs/threads";

FILE                                                    *logtpcc
                                = NULL;
FILE                                                    *errtpcc
                                = NULL;
FILE                                                    *comlog
                                = NULL;

pthread_mutex_t                 init_lock;

thread_info_t                   **info_list                                     =
NULL;

total_tran_count_t      total_counts;
total_tran_count_t      *pClientInfo                    = NULL;

tux_status_t                    tux_status;

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

/*
*****************************************************************
** Name                                 : get_environment_value
** Description:
**                                                      Get environment value
** Parameters :
**                                                      None
** Returns              :
**                                                      char * - environment name
**                                                      int - default value
** Comments             :
**
*****************************************************************
*/

int get_environment_value(char *name,int defaultValue)
{
        char * envValue = getenv(name);
        if(envValue)
                return atoi(envValue);

        return defaultValue;
}

/*
*****************************************************************
** Name                                 : get_thread_id
** Description:
**                                                      Get user thread id
** Parameters :
**                                                      None
** Returns              :
**                                                      int - thread id
** Comments             :
**
*****************************************************************
*/

int get_thread_id()
{
        int sequence            = 0;

        pthread_t thread = pthread_self();
        pthread_getunique_np(&thread,&sequence);

        return sequence;
}

/*
*****************************************************************
** Name                                 : time_diff_ms
** Description:
**                                                      Calculate time difference in ms
** Parameters :
**                                                      timeval * - end time
**                                                      timeval * - start time
** Returns              :
**                                                      int - time difference
** Comments             :
**
*****************************************************************
*/

int time_diff_ms(timeval *t2,timeval *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);
}
```

```
/*
*****************************************************************
** Name                                 : getUserSysTime
** Description:
**                                                      Calculate time usage
** Parameters :
**                                                      timeval * - User time
**                                                      timeval * - System time
** Returns              :
**                                                      None
** Comments             :
**
*****************************************************************
*/

void getUserSysTime(struct timeval *user_time, struct timeval *sys_time)
{
        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                       = 0;
                user_time->tv_usec      = 0;

                sys_time->tv_sec                        = 0;
                sys_time->tv_usec                       = 0;
        }

        return;
}

/*
*****************************************************************
** Name                                 : get_prefix
** Description:
**                                                      Format the output prefix for
printing
** Parameters :
**                                                      char * - buffer
** Returns              :
**                                                      None
** Comments             :
**
*****************************************************************
*/

void get_prefix(char *buffer)
{
        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 ? " " : "");

        return;
}

/*
*****************************************************************
** Name                                 : err_printf
** Description:
**                                                      Write error record
** Parameters :
**                                                      char * - message
** Returns              :
**                                                      None
** Comments             :
**
*****************************************************************
*/

void err_printf(char *logmsg)
{
        #if ERROR_LOG
                time_t          cur_timet;

                char                    time_str[30];
                char                    line_prefix[50];

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

                get_prefix(line_prefix);
                fprintf(ERROUT,"%s %s - %s\n",line_prefix,time_str,logmsg);
        #endif

        return;
```

```
}

/*
*****************************************************************
** Name                          : logprintf
** Description:
**                                                   Write log record
** Parameters :
**                                                   char * - format
**                                                   ? - variable arguments
** Returns                 :
**                                                   None
** Comments                :
**
*****************************************************************
*/

void logprintf(char *logmsg)
{
        #if DEBUG_LOG
                time_t          cur_timet;

                char                    time_str[30];
                char                    line_prefix[50];

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

                get_prefix(line_prefix);
                fprintf(logtpcc,"%s %s - %s\n",line_prefix,time_str,logmsg);

                if (logtpcc)
                        fflush(logtpcc);
        #endif

        return;
}

/*
*****************************************************************
** Name                          : prefix_sprintf
** Description:
**                                                   Write prefix record
** Parameters :
**                                                   char * - buffer
**                                                   char * - format
**                                                   ? - variable arguments
** Returns                 :
**                                                   None
** Comments                :
**
*****************************************************************
*/

void prefix_sprintf(char *buf,char *logmsg)
{
        time_t          cur_timet;

        char                    time_str[30];
        char                    line_prefix[50];
        char                    info[256];

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

        get_prefix(line_prefix);
        sprintf(buf,"%s %s - %s",line_prefix,time_str,logmsg);

        strcat(buf,info);

        return;
}

/*
*****************************************************************
** Name                          : write_coms_entry
** Description:
**                                                   Write a record to the comms
log
** Parameters :
**                                                   char * - comms data
**                                                   char - Input/Output
** Returns                 :
**                                                   None
** Comments                :
**
*****************************************************************
*/

void write_comms_entry(char *comms_data,char operation)
{
        int             pos                                     = 0;
        char            buffer[16384];
        char            data;

        fprintf(comlog,"%c: ",operation);

        do
        {
                data = *comms_data++;
                if(data)
                {
                        switch(data)
                        {
                        case 27:
                                buffer[pos++] = '<';
                                buffer[pos++] = 'e';
                                buffer[pos++] = 's';
                                buffer[pos++] = 'c';
                                buffer[pos++] = '>';
                                break;
                        case '\t':
                                buffer[pos++] = '<';
                                buffer[pos++] = 't';
                                buffer[pos++] = 'a';
                                buffer[pos++] = 'b';
                                buffer[pos++] = '>';
                                break;
                        case '\n':
                                buffer[pos++] = '<';
                                buffer[pos++] = 'n';
                                buffer[pos++] = 'l';
                                buffer[pos++] = '>';
                                break;
                        case '\r':
                                buffer[pos++] = '<';
                                buffer[pos++] = 'c';
                                buffer[pos++] = 'r';
                                buffer[pos++] = '>';
                                break;
                        default:
                                buffer[pos++] = data;
                                break;
                        }
                }
        } while(data);

        buffer[pos] = NULL;

        fprintf(comlog,"%s\n",buffer); fflush(comlog);

        return;
}


/*
*****************************************************************
** Name                          : format_char
** Description:
**                                                   Format a character string
** Parameters :
**                                                   char * - buffer
**                                                   int - size
**                                                   char - value
** Returns                 :
**                                                   int - buffer pos
** Comments                :
**
*****************************************************************
*/

int format_char(char *buf, int size, char val)
{
        int neg;
        int pos         = size;

        buf[--pos] = NULL;

        // format value
        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 ERR;
        }

        // Negitive?
        if(neg)
                buf[--pos] = '-';

        // Add space
```

```
		if(pos)
			memset(buf,' ',pos);

		// return buffer pos
		return pos;
}

/*
*****************************************************************
** Name				: format_short
** Description:
**					Format a short
** Parameters :
**					char * - buffer
**					int - size
**					short - value
** Returns		:
**					int - buffer pos
** Comments		:
**
*****************************************************************
*/

int format_short(char *buf,int size,short val)
{
		int neg;
		int pos			= size;

		buf[--pos] = NULL;

		// format value
		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 ERR;
		}

		// Negitive?
		if(neg)
			buf[--pos] = '-';

		// Add space
		if(pos)
			memset(buf,' ',pos);

		// Return buffer pos
		return pos;
}

/*
*****************************************************************
** Name				: format_int
** Description:
**					Format a int
** Parameters :
**					char * - buffer
**					int - size
**					int - value
** Returns		:
**					int - buffer pos
** Comments		:
**
*****************************************************************
*/

int format_int(char *buf,int size,int val)
{
		int		neg;
		int		pos					= size;

		buf[--pos] = 0;

		// format value
		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 ERR;
		}

		// Negitive?
		if(neg)
			buf[--pos] = '-';

		// Add space
		if (pos)
			memset(buf,' ',pos);

		// Return buffer pos
		return pos;
}

/*
*****************************************************************
** Name				: format_long
** Description:
**					Format a long
** Parameters :
**					char * - buffer
**					int - size
**					long - value
** Returns		:
**					int - buffer pos
** Comments		:
**
*****************************************************************
*/

int format_long(char *buf, int size,long val)
{
		int		neg;
		int		pos				= size;

		buf[--pos] = NULL;

		// format value
		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 ERR;
		}

		// Negitive?
		if(neg)
			buf[--pos] = '-';

		// Add space
		if (pos)
			memset(buf,' ',pos);

		// Return buffer pos
		return pos;
}

/*
*****************************************************************
** Name				: format_float
** Description:
**					Format a float
** Parameters :
**					char * - buffer
**					int - size
**					int - decimal prec.
**					float - value
** Returns		:
```

```
**                                                              int - buffer pos
** Comments                :
**
*****************************************************************
*/

int format_float(char *buf,int size,int dec,double val)
{
        static double   pow10[]         = {1,10,100,1000,10000,100000,1000000 };

        int                                                     neg;
        int                                                     pos                     =
size;

        buf[--pos] = NULL;

        val             = rint(val * pow10[dec]);
        neg             = (val < 0) ? 1 : 0;

        if(neg) val = -val;

        // Format value
        while(val >= 1 && pos > 0)
        {
                if(!dec--)
                {
                        buf[--pos] = '.';
                        continue;
                }

                buf[--pos] = (int) fmod(val, 10) + '0';
                val /= 10;
        }

        // Format decimal
        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 ERR;
        }

        // Negitive?
        if(neg)
                buf[--pos] = '-';

        // Add space
        if(pos)
                memset(buf,' ',pos);

        // Return buffer pos
        return pos;
}

/*
*****************************************************************
** Name                                 : format_money
** Description:
**                                                              Format money
** Parameters :
**                                                              char * - buffer
**                                                              int - size
**                                                              double - value
** Returns                 :
**                                                              int - buffer pos
** Comments                :
**
*****************************************************************
*/

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;
}

/*
*****************************************************************
** Name                                 : format_date
** Description:
**                                                              Format date
** Parameters :
**                                                              char * - buffer
**                                                              int - size
**                                                              char * - value
** Returns                 :
**                                                              Zero
** Comments                :
**
*****************************************************************
*/

int format_date(char *buf,int size,char *val)
{
        memcpy(buf,val,size);
        buf[size] = NULL;

        return 0;
}

/*
*****************************************************************
** Name                                 : format_date
** Description:
**                                                              Format date
** Parameters :
**                                                              char * - buffer
**                                                              int - size
**                                                              int64 * - value
** Returns                 :
**                                                              Zero
** Comments                :
**
*****************************************************************
*/

int format_date(char *buf,int size,int64 val)
{
        int                                     date                    = (int) (val &
0xffffffff);
        int                                     offset          = (int) (val >> 32);

        struct tm       dateTime;

        memcpy((void *) &dateTime,localtime(&date),sizeof(struct tm));
        dateTime.tm_year += offset;

        strftime(buf,32,"%m-%e-%Y",&dateTime);
        return 0;
}


/*
*****************************************************************
** Name                                 : format_datetime
** Description:
**                                                              Format date & time
** Parameters :
**                                                              char * - buffer
**                                                              int - size
**                                                              int64 * - value
** Returns                 :
**                                                              Zero
** Comments                :
**
*****************************************************************
*/

int format_datetime(char *buf,int size,int64 val)
{
        int                                     date                    = (int) (val &
0xffffffff);
        int                                     offset          = (int) (val >> 32);

        struct tm       dateTime;

        memcpy((void *) &dateTime,localtime(&date),sizeof(struct tm));
        dateTime.tm_year += offset;

        strftime(buf,32,"%m-%e-%Y %H:%M:%S",&dateTime);
        return 0;
}

/*
*****************************************************************
** Name                                 : format_phone
** Description:
**                                                              Format phone number
** Parameters :
**                                                              char * - buffer
**                                                              int - size
**                                                              char * - value
** Returns                 :
**                                                              Buffer pos
** Comments                :
**
*****************************************************************
*/

int format_phone(char *buf,int size,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;
}

/*
*****************************************************************
** Name                        : format_zip
** Description:
**                                                Format zip code
** Parameters :
**                                                char * - buffer
**                                                int - size
**                                                char * - value
** Returns                :
**                                                Buffer pos
** Comments               :
**
*****************************************************************
*/

int format_zip(char *buf,int size,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;
}

/*
*****************************************************************
** Name                        : string_empty
** Description:
**                                                Check if string empty
** Parameters :
**                                                char * - buffer
** Returns                :
**                                                int - Result
** Comments               :
**
*****************************************************************
*/

int string_empty(char *data)
{
        return data[0] == 0;
}

/*
*****************************************************************
** Name                        : pos_zero
** Description:
**                                                Check if value is zero
** Parameters :
**                                                int * - value
** Returns                :
**                                                int - Result
** Comments               :
**
*****************************************************************
*/

int pos_zero(int *val)
{
        return *val == 0;
}

/*
*****************************************************************
** Name                        : pos_nonzeros
** Description:
**                                                Check if values are non-zero
** Parameters :
**                                                int ** - values
** Returns                :
**                                                int - Result
** Comments               :
**
*****************************************************************
*/

int pos_nonzeros(int **val)
{
        for (int **ptr = val;*ptr;ptr++)
        {
                if (**ptr == 0)
                        return 0;
        }

        return 1;
}

/*
*****************************************************************
** Name                        : isdigit
** Description:
**                                                Check if values are digits
** Parameters :
**                                                int - values
** Returns                :
**                                                int - TRUE = yes, FALSE = no
** Comments               :
**
*****************************************************************
*/

int isdigit(int value)
{
        if(value != '-' && value != '.')
                if(value < '0' || value > '9')
                        return FALSE;

        return TRUE;
}

/*
*****************************************************************
** Name                        : isprint
** Description:
**                                                Check if values are printable
chars?
** Parameters :
**                                                int - values
** Returns                :
**                                                int - TRUE = yes, FALSE = no
** Comments               :
**
*****************************************************************
*/

int isprint(int value)
{
        if(value >= '0' && value <= '9')
                return TRUE;

        if(value >= 'a' && value <= 'z')
                return TRUE;

        if(value >= 'A' && value <= 'Z')
                return TRUE;

        switch(value)
        {
        case '.':
        case ',':
        case '$':
        case ' ':
        case '-':
                return TRUE;
        }

        return FALSE;
}

//////////////////////////////////////////////////////////////////////
```

tuxClientCommon.hpp

```
/*
*****************************************************************
** Project                : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                        : tuxClientCommon.hpp
** Title                       : Tux client common helpers
*****************************************************************
```

```
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History       :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                                    :
**
*****************************************************************
*/

#ifndef __tuxClientCommon_hpp__
#define __tuxClientCommon_hpp__

#include <commonTuxDB2.hpp>
#include <tuxClientInf.hpp>

//////////////////////////////////////////////////////////////////
// Externals
//////////////////////////////////////////////////////////////////

extern int                                                  user_id;
extern int                                                  useSecurity;
extern int                                                  debug;
extern int                                                  info_list_len;
extern int                                                  warehouse_offset;
extern int                                                  logtrans;
extern int
client_lock_handles;
extern int                                                  user_port;
extern int
consecutiveErrors;
extern int                                                  iStatsFrequency;
extern int                                                  envRetrieval;
extern int
num_active_threads;

extern char                                                 host_name[256];

extern char
log_file_name[256];
extern char
err_file_name[256];
extern char
com_file_name[256];

extern char                                                 *user_code;
//extern char                                               **sys_errlist;
extern char                                                 *result_dir;
extern char
*LOG_FILE_DIR;

extern FILE                                                 *logtpcc;
extern FILE                                                 *errtpcc;
extern FILE                                                 *comlog;

extern pthread_mutex_t                  init_lock;

extern thread_info_t                            **info_list;

extern total_tran_count_t       total_counts;
extern total_tran_count_t       *pClientInfo;

extern tux_status_t                             tux_status;

//////////////////////////////////////////////////////////////////
// Definitions
//////////////////////////////////////////////////////////////////

#define ERROUT                                          errtpcc
#define MAX_MSG_LEN                             4096

//////////////////////////////////////////////////////////////////
// Macros
//////////////////////////////////////////////////////////////////

#define CHECK_ENVIRON(str,var)
                                                                                \
                if (str == NULL)

                                        \
                {

                                                                                \
                fprintf(ERROUT,"%s environment variable is not defined.\n",var);        \
                }

//////////////////////////////////////////////////////////////////
// Function Definitions
//////////////////////////////////////////////////////////////////

int get_thread_id();
int get_environment_value(char *name,int defaultValue);

int time_diff_ms(timeval *t2,timeval *t1);
void getUserSysTime(struct timeval *user_time, struct timeval *sys_time);

void logprintf(char *logmsg);
void err_printf(char *logmsg);
void prefix_sprintf(char *buf,char *logmsg);
void get_prefix(char *buffer);
void write_comms_entry(char *comms_data,char operation);

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,char *val);
int format_date (char *buf,int size,int64 val);
int format_datetime (char *buf,int size,int64 val);
int format_phone(char *buf,int size,char *phone);
int format_zip  (char *buf,int size,char *zip);

int string_empty(char *data);
int pos_zero(int *val);
int pos_nonzeros(int **val);

int isdigit(int value);
int isprint(int value);

#endif
//////////////////////////////////////////////////////////////////
```

tuxClientField.cpp

```
/*
*****************************************************************
** Project                      : AIX DB/2 TPC-C
** Component : TPC-C/Client
** Name                                    : tuxClientField.cpp
** Title                                   : Tux client Field type
*****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History       :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                                    :
**
*****************************************************************
*/

#include <tuxClientField.hpp>

//////////////////////////////////////////////////////////////////
// Field class
//////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name       : Field::Field
** Description :
**              Constructor
** Parameters  :
**              Terminal * - device
**                                                                      int - size
**                                                                      char * - name
** Returns     :
**              None
** Comments    :
**
*****************************************************************
*/

Field::Field(Terminal *inoutP,int size,char *str)
                                                        : ioP(inoutP), len(size), pos(0), changed(0),
need_redisplay(0)
{
                need_free                               = 0;
                need_free_string                = 0;
                dependent                               = 0;

                if(str == NULL)
                {
                                string = new char[len + 1];
                                need_free_string = 1;
                }
                else string = str;

                ok_func                         = NULL;
                ok_data                         = NULL;
                *string                         = NULL;

                return;
}

/*
*****************************************************************
** Name       : Field::Field
```

```
** Description :
**          Constructor
** Parameters  :
**          Terminal * - device
**                                                          int - x pos
**                                                          int y pos
**                                                          int - size
**                                                          char * - name
** Returns     :
**          None
** Comments    :
**
*****************************************************************
*/

Field::Field(Terminal *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                           = 0;
            need_free_string        = 0;
            dependent                           = 0;

            if(str == NULL)
            {
                        string = new char[len+1];
                        need_free_string = 1;
            }
            else string = str;

            ok_func                     = NULL;
            ok_data                     = NULL;
            *string                     = NULL;
}

/*
*****************************************************************
** Name       : Field::reset
** Description :
**          Reset field
** Parameters  :
**          None
** Returns     :
**          int - return code
** Comments    :
**
*****************************************************************
*/

int Field::reset()
{
            pos                         = 0;
            changed     = 0;

            return 0;
}

/*
*****************************************************************
** Name       : Field::finalize_field
** Description :
**          Value for field entered
** Parameters  :
**          None
** Returns     :
**          int - return code
** Comments    :
**
*****************************************************************
*/

int Field::finalize_field()
{
            changed                     = 0;
            string[pos]     = 0;

            return 0;
}

/*
*****************************************************************
** Name       : Field::display_field
** Description :
**          Display field for entry
** Parameters  :
**          int - underscore flag
** Returns     :
**          int - return code
** Comments    :
**
*****************************************************************
*/

int Field::display_field(int use_underscores)
{
            // position cursor
            position(ioP,x,y);

            // write field name
            ioP->write(string);

            // mark field
            if(use_underscores)
                        ioP->write(underscores,len-pos);
            else ioP->write(blanks,len-pos);

            return 0;
}

/*
*****************************************************************
** Name       : Field::get_key
** Description :
**          Key entry
** Parameters  :
**          None
** Returns     :
**          int - return code
** Comments    :
**
*****************************************************************
*/

int Field::get_key()
{
            char        key;
            int         rc;

            rc = ioP->read(&key, 1);
            return (rc == 0)  ? EOF : key ;
}

/*
*****************************************************************
** Name       : Field::add_char
** Description :
**          Field add
** Parameters  :
**          int - key stroke
** Returns     :
**          int - return code
** Comments    :
**
*****************************************************************
*/

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;
}

/*
*****************************************************************
** Name       : Field::backspace
** Description :
**          Field backspace
** Parameters  :
**          None
** Returns     :
**          int - return code
** Comments    :
**
*****************************************************************
*/

int Field::backspace()
{
            ioP->write("\b_\b", 3);

            changed = 1; pos--;
            return 0;
}

/*
*****************************************************************
** Name       : Field::start_position
** Description :
**          Field set start position
** Parameters  :
**          None
** Returns     :
**          int - return code
** Comments    :
**
*****************************************************************
*/

int Field::start_position ()
```

```
{
        position(ioP,x,y);
        return 0;
}

/*
*****************************************************************
** Name       : Field::get_field
** Description :
**         Field entry
** Parameters  :
**         None
** Returns    :
**         int - return code
** Comments   :
**                                                         process each key stroke
**
*****************************************************************
*/

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;
                case '\t':
                        // Tab
                case '\006':
                // Ctrl-F
                case '\016':
                // Ctrl-N
                        if(changed)
                                finalize_field();

                        ioP->pop();

                        display_field(1);
                        return NEXT_FIELD;
                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);
                }
        }

        return 0;
}

/*
*****************************************************************
** Name       : Field::~Field
** Description :
**         Destructor
** Parameters  :
**         None
** Returns    :
**         None
** Comments   :
**
*****************************************************************
*/

Field::~Field()
{
        if(need_free_string)
                delete [] string;

        return;
}

/*
*****************************************************************
** Name       : Field::update_field_address
** Description :
**         Update address of value *
** Parameters  :
**         int - number of byte +/- to adjust value *
** Returns    :
**         None
** Comments   :
**                                                         As the buffer address of the
data structure
**                                                         which these fields use can
change we need to
**                                                         adjust the value * when this
change occurs
**
**                                                         This only occurs if the value ptr
is managed
**                                                         outside of the class. I.E. the
value pointer
**                                                         was passed in. Therfore we
check need_free to
**                                                         see if we are manageing it
withing the class.
**
*****************************************************************
*/

void Field::update_field_address(int adjustment)
{
        if(need_free_string) return;

        char *oldptr = string;
        string += adjustment;
        char *newptr = string;
        return;
}

////////////////////////////////////////////////////////////////////
// IntField class
////////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name       : IntField::IntField
** Description :
**         Constructor
** Parameters  :
**         Terminal * - input/output
**                                                         int - xpos
**                                                         int - ypos
**                                                         int - size
**                                                         char * - name
** Returns    :
**         None
** Comments   :
**
*****************************************************************
*/

IntField::IntField(Terminal *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;
                }

                return;
}

/*
*********************************************************************
** Name       : IntField::IntField
** Description :
**           Constructor
** Parameters  :
**           Terminal * - input/output
**                                                          int - size
**                                                          char * - name
** Returns     :
**           None
** Comments    :
**
*********************************************************************
*/

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

                return;
}

/*
*********************************************************************
** Name       : IntField::~IntField
** Description :
**           Destructor
** Parameters  :
**           None
** Returns     :
**           None
** Comments    :
**
*********************************************************************
*/

IntField::~IntField()
{
                if(need_free)
                        delete value;

                return;
}

/*
*********************************************************************
** Name       : IntField::add_char
** Description :
**           Add a character
** Parameters  :
**           int - key stroke
** Returns     :
**           None
** Comments    :
**
*********************************************************************
*/

int IntField::add_char(int key)
{
                // check numeric key
                if(pos < len && isdigit(key))
                {
                        changed = 1;
                        string[pos] = key;
                        ioP->write(&string[pos++],1);
                        return 0;
                }

                ioP->write("\a", 1);
                return 1;
}

/*
*********************************************************************
** Name       : IntField::display_field
** Description :
**           Display the field for entry
** Parameters  :
**           int - Use underscores
** Returns     :
**           None
** Comments    :
**
*********************************************************************
*/

int IntField::display_field(int use_underscores)
{
                int             firstchar;
                char            *buf                                    = new char[len + 1];

                // clear buffer
                memset(buf,'x',len);

                if(pos)
                        firstchar = format_int(buf,len + 1,*value);
                else firstchar = len;

                position(ioP, x, y);

                // display field
                if(use_underscores)
                {
                        ioP->write(underscores,firstchar);
                        ioP->write(buf + firstchar,len - firstchar);
                }
                else ioP->write(buf,len);

                return 0;
}

/*
*********************************************************************
** Name       : IntField::finalize_field
** Description :
**           Value for field entered
** Parameters  :
**           None
** Returns     :
**           int - return code
** Comments    :
**
*********************************************************************
*/

int IntField::finalize_field()
{
                changed                 = 0;
                string[pos]     = 0;

                if (value != NULL)
                        *value = atoi(string);

                return 0;
}

/*
*********************************************************************
** Name       : IntField::update_field_address
** Description :
**           Update address of value *
** Parameters  :
**           int - number of byte +/- to adjust value *
** Returns     :
**           None
** Comments    :
**                                                          As the buffer address of the
data structure
**                                                          which these fields use can
change we need to
**                                                          adjust the value * when this
change occurs
**
**                                                          This only occurs if the value ptr
is managed
**                                                          outside of the class. I.E. the
value pointer
**                                                          was passed in. Therfore we
check need_free to
**                                                          see if we are manageing it
withing the class.
**
*********************************************************************
*/

void IntField::update_field_address(int adjustment)
{
                if(need_free) return;

                int *oldptr = value;
                value = (int *) (((char *) value) + adjustment);
                int *newptr = value;
                return;
}

//////////////////////////////////////////////////////////////////////
// ShortField
//////////////////////////////////////////////////////////////////////

/*
*********************************************************************
** Name       : ShortField::ShortField
** Description :
```

```
**          Constructor
** Parameters  :
**          Terminal * - input/output
**                                                          int - xpos
**                                                          int - ypos
**                                                          int - size
**                                                          char * - name
** Returns     :
**          None
** Comments    :
**
*****************************************************************
*/

ShortField::ShortField(Terminal *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;
        }

        return;
}

/*
*****************************************************************
** Name       : ShortField::ShortField
** Description :
**          Constructor
** Parameters  :
**          Terminal * - input/output
**                                                          int - size
**                                                          char * - name
** Returns     :
**          None
** Comments    :
**
*****************************************************************
*/

ShortField::ShortField(Terminal *ioP,int size,short *val) : Field(ioP,size),value(val)
{
        if(value == NULL)
        {
                value = new short;
                need_free = 1;
        }

        return;
}

/*
*****************************************************************
** Name       : ShortField::~ShortField
** Description :
**          Destructor
** Parameters  :
**          None
** Returns     :
**          None
** Comments    :
**
*****************************************************************
*/

ShortField::~ShortField()
{
        if(need_free)
                delete value;

        return;
}

/*
*****************************************************************
** Name       : ShortField::add_char
** Description :
**          Add a character
** Parameters  :
**          int - key stroke
** Returns     :
**          None
** Comments    :
**
*****************************************************************
*/

int ShortField::add_char(int key)
{
        // check numeric key
        if(pos < len && isdigit(key))
        {
                changed = 1;
                string[pos] = key;
                ioP->write(&string[pos++],1);
                return 0;
        }

        ioP->write("\a", 1);
        return 1;
}

/*
*****************************************************************
** Name       : ShortField::display_field
** Description :
**          Display the field for entry
** Parameters  :
**          int - Use underscores
** Returns     :
**          None
** Comments    :
**
*****************************************************************
*/

int ShortField::display_field(int use_underscores)
{
        int             firstchar;
        char            *buf                                    = new char[len + 1];

        if(pos)
                firstchar = format_short(buf,len + 1,*value);
        else firstchar = len;

        position(ioP,x,y);

        // Display field
        if(use_underscores)
        {
                ioP->write(underscores,firstchar);
                ioP->write(buf + firstchar,len - firstchar);
        }
        else ioP->write(buf, len);

        return 0;
}

/*
*****************************************************************
** Name       : ShortField::finalize_field
** Description :
**          Value for field entered
** Parameters  :
**          None
** Returns     :
**          int - return code
** Comments    :
**
*****************************************************************
*/

int ShortField::finalize_field()
{
        changed                 = 0;
        string[pos]     = 0;

        if (value != NULL)
                *value = atoi(string);

        return 0;
}

/*
*****************************************************************
** Name       : ShortField::update_field_address
** Description :
**          Update address of value *
** Parameters  :
**          int - number of byte +/- to adjust value *
** Returns     :
**          None
** Comments    :
**                                                          As the buffer address of the
data structure
**                                                          which these fields use can
change we need to
**                                                          adjust the value * when this
change occurs
**
**                                                          This only occurs if the value ptr
is managed
**                                                          outside of the class. I.E. the
value pointer
**                                                          was passed in. Therfore we
check need_free to
**                                                          see if we are manageing it
withing the class.
**
*****************************************************************
*/

void ShortField::update_field_address(int adjustment)
{
        if(need_free) return;
```

```
                short *oldptr = value;
                value = (short *) (((char *) value) + adjustment);
                short *newptr = value;
                return;
}

////////////////////////////////////////////////////////////////////
// Int8Field class
////////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name       : Int8Field::Int8Field
** Description :
**             Constructor
** Parameters  :
**             Terminal * - input/output
**                                                          int - xpos
**                                                          int - ypos
**                                                          int - size
**                                                          unsigned char * - name
** Returns     :
**             None
** Comments    :
**
*****************************************************************
*/

Int8Field::Int8Field(Terminal *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;
                }

                return;
}

/*
*****************************************************************
** Name       : Int8Field::Int8Field
** Description :
**             Constructor
** Parameters  :
**             Terminal * - input/output
**                                                          int - size
**                                                          unsigned char * - name
** Returns     :
**             None
** Comments    :
**
*****************************************************************
*/

Int8Field::Int8Field(Terminal *ioP,int size,unsigned char *val) : Field(ioP,size),value(val)
{
                if(value == NULL)
                {
                                value = new unsigned char;
                                need_free = 1;
                }

                return;
}

/*
*****************************************************************
** Name       : Int8Field::~Int8Field
** Description :
**             Destructor
** Parameters  :
**             None
** Returns     :
**             None
** Comments    :
**
*****************************************************************
*/

Int8Field::~Int8Field()
{
                if(need_free)
                                delete value;

                return;
}

/*
*****************************************************************
** Name       : Int8Field::add_char
** Description :
**             Add a character
** Parameters  :
**             int - key stroke
** Returns     :
**             None
** Comments    :
**
*****************************************************************
*/

int Int8Field::add_char(int key)
{
                // check if digit key pressed
                if(pos < len && isdigit(key))
                {
                                changed = 1;
                                string[pos] = key;
                                ioP->write(&string[pos++], 1);
                                return 0;
                }

                ioP->write("\a",1);
                return 1;
}

/*
*****************************************************************
** Name       : Int8Field::display_field
** Description :
**             Display the field for entry
** Parameters  :
**             int - Use underscores
** Returns     :
**             None
** Comments    :
**
*****************************************************************
*/

int Int8Field::display_field(int use_underscores)
{
                int             firstchar;
                char            *buf                                    = new char[len + 1];

                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;
}

/*
*****************************************************************
** Name       : Int8Field::finalize_field
** Description :
**             Value for field entered
** Parameters  :
**             None
** Returns     :
**             int - return code
** Comments    :
**
*****************************************************************
*/

int Int8Field::finalize_field()
{
                changed                         = 0;
                string[pos]     = 0;

                if(value != NULL)
                                *value = atoi(string);

                return 0;
}

/*
*****************************************************************
** Name       : Int8Field::update_field_address
** Description :
**             Update address of value *
** Parameters  :
**             int - number of byte +/- to adjust value *
** Returns     :
**             None
** Comments    :
**                                                          As the buffer address of the
data structure
**                                                          which these fields use can
change we need to
**                                                          adjust the value * when this
change occurs
**
**                                                          This only occurs if the value ptr
```

```
is managed
**                                                              outside of the class. I.E. the
value pointer
**                                                              was passed in. Therfore we
check need_free to
**                                                              see if we are manageing it
withing the class.
**
*****************************************************************
*/

void Int8Field::update_field_address(int adjustment)
{
        if(need_free) return;

        unsigned char *oldptr = value;
        value = (unsigned char *) (((char *) value) + adjustment);
        unsigned char *newptr = value;
        return;
}

////////////////////////////////////////////////////////////////////////
// LongField class
////////////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name       : LongField::LongField
** Description :
**           Constructor
** Parameters  :
**           Terminal * - input/output
**                                                              int - xpos
**                                                              int - ypos
**                                                              int - size
**                                                              unsigned char * - name
** Returns    :
**           None
** Comments   :
**
*****************************************************************
*/

LongField::LongField(Terminal *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;
        }

        return;
}

/*
*****************************************************************
** Name       : LongField::LongField
** Description :
**           Constructor
** Parameters  :
**           Terminal * - input/output
**                                                              int - size
**                                                              unsigned char * - name
** Returns    :
**           None
** Comments   :
**
*****************************************************************
*/

LongField::LongField(Terminal *ioP,int size,long *val) : Field(ioP,size),value(val)
{
        if(value == NULL)
        {
                value = new long;
                need_free = 1;
        }

        return;
}

/*
*****************************************************************
** Name       : LongField::~LongField
** Description :
**           Destructor
** Parameters  :
**           None
** Returns    :
**           None
** Comments   :
**
*****************************************************************
*/

LongField::~LongField()
{
        if(need_free)
                delete value;

        return;
}

/*
*****************************************************************
** Name       : longField::add_char
** Description :
**           Add a character
** Parameters  :
**           int - key stroke
** Returns    :
**           None
** Comments   :
**
*****************************************************************
*/

int LongField::add_char(int key)
{
        // check digit key pressed
        if(pos < len && isdigit(key))
        {
                changed = 1;
                string[pos] = key;
                ioP->write(&string[pos++],1);
                return 0;
        }

        ioP->write("\a",1);
        return 1;
}

/*
*****************************************************************
** Name       : LongField::display_field
** Description :
**           Display the field for entry
** Parameters  :
**           int - Use underscores
** Returns    :
**           None
** Comments   :
**
*****************************************************************
*/

int LongField::display_field(int use_underscores)
{
        int             firstchar;
        char            *buf                                    = new char[len + 1];

        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;
}

/*
*****************************************************************
** Name       : LongField::finalize_field
** Description :
**           Value for field entered
** Parameters  :
**           None
** Returns    :
**           int - return code
** Comments   :
**
*****************************************************************
*/

int LongField::finalize_field()
{
        changed                 = 0;
        string[pos]     = 0;

        if(value != NULL)
                *value = atoi(string);

        return 0;
}

/*
*****************************************************************
** Name       : LongField::update_field_address
** Description :
```

```
**              Update address of value *
** Parameters  :
**              int - number of byte +/- to adjust value *
** Returns     :
**              None
** Comments    :
**                                                                      As the buffer address of the
data structure
**                                                                      which these fields use can
change we need to
**                                                                      adjust the value * when this
change occurs
**
**                                                                      This only occurs if the value ptr
is managed
**                                                                      outside of the class. I.E. the
value pointer
**                                                                      was passed in. Therfore we
check need_free to
**                                                                      see if we are manageing it
withing the class.
**
*****************************************************************
*/

void LongField::update_field_address(int adjustment)
{
                if(need_free) return;

                long *oldptr = value;
                value = (long *) (((char *) value) + adjustment);
                long *newptr = value;
                return;
}

////////////////////////////////////////////////////////////////////////
// MoneyField class
////////////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name       : MoneyField::MoneyField
** Description :
**              Constructor
** Parameters  :
**              Terminal * - input/output
**                                                                      int - xpos
**                                                                      int - ypos
**                                                                      int - size
**                                                                      unsigned char * - name
** Returns     :
**              None
** Comments    :
**
*****************************************************************
*/

MoneyField::MoneyField(Terminal *ioP,int inx,int iny,int size,int64_t *val)
                                                                                                        :
Field(ioP,inx,iny,size),value(val)
{
                seen_dollar     = 0; seen_sign = 0;
                seen_dot                = 0; seen_digit= 0;

                if(value == NULL)
                {
                                value = new int64_t;
                                need_free = 1;
                }

                return;
}

/*
*****************************************************************
** Name       : MoneyField::MoneyField
** Description :
**              Constructor
** Parameters  :
**              Terminal * - input/output
**                                                                      int - size
**                                                                      unsigned char * - name
** Returns     :
**              None
** Comments    :
**
*****************************************************************
*/

MoneyField::MoneyField(Terminal *ioP,int size,int64 *val) : Field(ioP,size),value(val)
{
                seen_dollar     = 0; seen_sign = 0;
                seen_dot                = 0; seen_digit= 0;

                if(value == NULL)
                {
                                value = new int64_t;
                                need_free = 1;
                }
                return;
}

/*
*****************************************************************
** Name       : MoneyField::~MoneyField
** Description :
**              Destructor
** Parameters  :
**                                                                                                      None
** Returns     :
**              None
** Comments    :
**
*****************************************************************
*/

MoneyField::~MoneyField()
{
                if(need_free)
                                delete value;

                return;
}

/*
*****************************************************************
** Name       : MoneyField::add_char
** Description :
**              Add a character
** Parameters  :
**              int - key stroke
** Returns     :
**              None
** Comments    :
**
*****************************************************************
*/

int MoneyField::add_char(int key)
{
                do
                {
                                if(pos >= len)
                                                break;

                                // Process special key
                                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;
                                                continue;
                                }
                                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;
}

/*
*****************************************************************
** Name       : MoneyField::backspace
** Description :
**              Field backspace
** Parameters  :
**              None
```

```
** Returns     :
**              int - return code
** Comments    :
**
*****************************************************************
*/

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;
}

/*
*****************************************************************
** Name       : MoneyField::display_field
** Description :
**              Display the field for entry
** Parameters  :
**              int - Use underscores
** Returns     :
**              None
** Comments    :
**
*****************************************************************
*/

int MoneyField::display_field(int use_underscores)
{
                int             firstchar;
                char            *buf                                    = new char[len + 1];

                if(pos)
                                firstchar = format_money(buf,len + 1,(double)((*value)/100.0));
                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;
}

/*
*****************************************************************
** Name       : MoneyField::finalize_field
** Description :
**              Value for field entered
** Parameters  :
**              None
** Returns     :
**              int - return code
** Comments    :
**
*****************************************************************
*/

int MoneyField::finalize_field()
{
                changed                 = 0;
                string[pos]     = 0;
                char logmsg[256];

                if(value != NULL)
                {
                                //*value = atof(string + seen_dollar + seen_sign);

                                *value=atoll(string + seen_dollar + seen_sign);

                                if(seen_sign)
                                                *value = -*value;
                }

                return 0;
}

/*
*****************************************************************
** Name       : MoneyField::reset
** Description :
**              Reset field
** Parameters  :
**              None
** Returns     :
**              int - return code
** Comments    :
**
*****************************************************************
*/

int MoneyField::reset()
{
                Field::reset();
                seen_dollar     = 0; seen_sign = 0;
                seen_dot                = 0; seen_digit= 0;

                return 0;
}

/*
*****************************************************************
** Name       : MoneyField::update_field_address
** Description :
**              Update address of value *
** Parameters  :
**              int - number of byte +/- to adjust value *
** Returns     :
**              None
** Comments    :
**                                                              As the buffer address of the
data structure
**                                                              which these fields use can
change we need to
**                                                              adjust the value * when this
change occurs
**
**                                                              This only occurs if the value ptr
is managed
**                                                              outside of the class. I.E. the
value pointer
**                                                              was passed in. Therfore we
check need_free to
**                                                              see if we are manageing it
withing the class.
**
*****************************************************************
*/

void MoneyField::update_field_address(int adjustment)
{
                if(need_free) return;

                int64_t *oldptr = value;
                value = (int64_t *) (((char *) value) + adjustment);
                int64_t *newptr = value;
                return;
}

//////////////////////////////////////////////////////////////////
// TextField class
//////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name       : TextField::TextField
** Description :
**              Constructor
** Parameters  :
**              Terminal * - input/output
**                                                              int - xpos
**                                                              int - ypos
**                                                              int - size
**                                                              char * - name
** Returns     :
**              None
** Comments    :
**
*****************************************************************
*/

TextField::TextField(Terminal *ioP,int inx,int iny,int size,char *str)
                                                                                                :
Field(ioP,inx,iny,size,str)
{
                value = TextField::string;
                return;
}

/*
*****************************************************************
** Name       : TextField::TextField
** Description :
**              Constructor
** Parameters  :
**              Terminal * - input/output
**                                                              int - xpos
**                                                              int - ypos
**                                                              int - size
```

```
**                                                              char * - name
** Returns    :
**             None
** Comments   :
**
*****************************************************************
*/

TextField::TextField(Terminal *ioP,int size,char *str) : Field(ioP,size,str)
{
                value = TextField::string;
                return;
}

/*
*****************************************************************
** Name       : TextField::add_char
** Description :
**             Add a character
** Parameters  :
**             int - key stroke
** Returns    :
**             None
** Comments   :
**
*****************************************************************
*/

int TextField::add_char(int key)
{
                changed                         = 1;
                string[pos]     = key;

                ioP->write(&string[pos++],1);
                return 0;
}

/*
*****************************************************************
** Name       : TextField::update_field_address
** Description :
**             Update address of value *
** Parameters  :
**             int - number of byte +/- to adjust value *
** Returns    :
**             None
** Comments   :
**                                                              As the buffer address of the
data structure
**                                                              which these fields use can
change we need to
**                                                              adjust the value * when this
change occurs
**
**                                                              This only occurs if the value ptr
is managed
**                                                              outside of the class. I.E. the
value pointer
**                                                              was passed in. Therfore we
check need_free to
**                                                              see if we are manageing it
withing the class.
**
*****************************************************************
*/

void TextField::update_field_address(int adjustment)
{
                if(need_free_string) return;

                char *oldptr = value;
                TextField::update_field_address(adjustment);
                value = TextField::string;
                char *newptr = value;

                return;
}

//////////////////////////////////////////////////////////////////////
// Utilities
//////////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name       : genfield
** Description :
**             Return a field pointer of type IntField
** Parameters  :
**             Terminal * - input/output
**                                                              int - xpos
**                                                              int - ypos
**                                                              int - size
**                                                              int * - value ptr
** Returns    :
**             None
** Comments   :
**
*****************************************************************
*/

Field *genfield(Terminal *ioP,int x,int y,int len,int *ptr)
{
                return new IntField(ioP,x, y,len,ptr);
}

/*
*****************************************************************
** Name       : genfield
** Description :
**             Return a field pointer of type ShortField
** Parameters  :
**             Terminal * - input/output
**                                                              int - xpos
**                                                              int - ypos
**                                                              int - size
**                                                              short * - value ptr
** Returns    :
**             None
** Comments   :
**
*****************************************************************
*/

Field *genfield(Terminal *ioP,int x,int y,int len,short *ptr)
{
                return new ShortField(ioP,x,y,len,ptr);
}

/*
*****************************************************************
** Name       : genfield
** Description :
**             Return a field pointer of type LongField
** Parameters  :
**             Terminal * - input/output
**                                                              int - xpos
**                                                              int - ypos
**                                                              int - size
**                                                              long * - value ptr
** Returns    :
**             None
** Comments   :
**
*****************************************************************
*/

Field *genfield(Terminal *ioP,int x,int y,int len,long *ptr)
{
                return new LongField(ioP,x,y,len,ptr);
}

/*
*****************************************************************
** Name       : genfield
** Description :
**             Return a field pointer of type TextField
** Parameters  :
**             Terminal * - input/output
**                                                              int - xpos
**                                                              int - ypos
**                                                              int - size
**                                                              char * - value ptr
** Returns    :
**             None
** Comments   :
**
*****************************************************************
*/

Field *genfield(Terminal *ioP,int x,int y,int len,char *ptr)
{
                return new TextField(ioP,x,y,len,ptr);
}

/*
*****************************************************************
** Name       : genfield
** Description :
**             Return a field pointer of type MoneyField
** Parameters  :
**             Terminal * - input/output
**                                                              int - xpos
**                                                              int - ypos
**                                                              int - size
**                                                              int64 * - value ptr
** Returns    :
**             None
** Comments   :
**
*****************************************************************
*/

Field *genfield(Terminal *ioP,int x,int y,int len,int64_t *ptr)
{
                return new MoneyField(ioP,x,y,len,ptr);
}

/*
```

```
/*****************************************************************
** Name       : genfield
** Description :
**             Return a field pointer of type Int8Field
** Parameters  :
**             Terminal * - input/output
**                                                          int - xpos
**                                                          int - ypos
**                                                          int - size
**                                                          unsigned char * - value ptr
** Returns     :
**             None
** Comments    :
**
*****************************************************************
*/

Field *genfield(Terminal *ioP,int x,int y,int len,unsigned char *ptr)
{
        return new Int8Field(ioP, x, y, len, ptr);
}

//////////////////////////////////////////////////////////////////
```

tuxClientField.hpp

```
/*
*****************************************************************
** Project                  : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                     : tuxClientField.hpp
** Title                    : Tux Field
*****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**             Developed at IBM Austin by the AIX RS/6000
**             performance group.
**
** Comments                 :
**
*****************************************************************
*/

#ifndef __tuxClientField_hpp__
#define __tuxClientField_hpp__

#include <commonTuxDB2.hpp>
#include <tuxClientCommon.hpp>

#include <tuxClientTerminal.hpp>

//////////////////////////////////////////////////////////////////
// Classes
//////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////
// Field class
//////////////////////////////////////////////////////////////////

class Field
{
        public:
                // embeded definitions
                enum return_codes
{INVALID,ENTER,NEXT_FIELD,PREV_FIELD,ABORT,REDISPLAY};
                class Error
                {
                        public:
                                enum {USER_ABORT};
                };

        // Constructors / Destructors
        public:
                                                        Field(Terminal *ioP,int size,char
*string = NULL);
                                                        Field(Terminal *ioP,int x,int
y,int size,char *string = NULL);

                virtual         ~Field();

        // Functions
        public:
                                                        int             get_key();
                                                        int
blank_with_prev()
        {return dependent;}
                                                        void
set_blank_with_prev(int enable = 1)             {dependent = enable; return;}

                virtual         int             reset();
                virtual         int             backspace();
                virtual         int             start_position();
                virtual         int             finalize_field();

                virtual         int             add_char(int key);

                virtual         int             get_field(int need_pos = 1);
                virtual         int             display_field(int use_underscores = 0);

                virtual         void            update_field_address(int adjustment);

        // Attributes
        public:
                Terminal        *ioP;

                int             x;
                int             y;
                int             len;
                int             pos;
                int             changed;
                int             need_free;
                int             need_redisplay;
                int             need_free_string;
                int             dependent;

                void            *ok_data;

                char            *string;

                int             (*ok_func)(void *data);
};

//////////////////////////////////////////////////////////////////
// Int8Field class
//////////////////////////////////////////////////////////////////

class Int8Field : public Field
{
        // Constructor / Destructor
        public:
                                                        Int8Field(Terminal *ioP,int x,int
y,int size,unsigned char *value = NULL);
                                                        Int8Field(Terminal *ioP,int
size,unsigned char *value = NULL);
                virtual         ~Int8Field();

        // Functions
        public:
                int             finalize_field();
                int             add_char(int key);
                int             display_field(int use_underscores = 0);

                void            update_field_address(int adjustment);

        // Attributes
        public:
                unsigned char *value;
};

//////////////////////////////////////////////////////////////////
// ShortField class
//////////////////////////////////////////////////////////////////

class ShortField : public Field
{
        // Constructor / Destructor
        public:
                                                        ShortField(Terminal *ioP,int
x,int y,int size,short *value = NULL);
                                                        ShortField(Terminal *ioP,int
size,short *value = NULL);
                virtual         ~ShortField();

        // Functions
        public:
                int finalize_field();
                int add_char(int key);
                int display_field(int use_underscores = 0);

                void            update_field_address(int adjustment);

        // Attributes
        public:
                short *value;

};

//////////////////////////////////////////////////////////////////
// IntField class
//////////////////////////////////////////////////////////////////

class IntField : public Field
{
        // Constructor / Destructor
        public:
                                                        IntField(Terminal *ioP,int x,int
y,int size,int *value = NULL);
                                                        IntField(Terminal *ioP,int
size,int *value = NULL);
                virtual         ~IntField();
```

```
		// Functions
		public:
			int finalize_field();
			int add_char(int key);
			int display_field(int use_underscores = 0);

			void		update_field_address(int adjustment);

		// Attributes
		public:
			int *value;
};

////////////////////////////////////////////////////////////////////
// LongField class
////////////////////////////////////////////////////////////////////

class LongField : public Field
{
		// Constructor / Destructor
		public:
							LongField(Terminal *ioP,int x,int y,int
size,long *value = NULL);
							LongField(Terminal *ioP,int size,long *value =
NULL);
		virtual		~LongField();

		// Functions
		public:
			int finalize_field();
			int add_char(int key);
			int display_field(int use_underscores = 0);

			void		update_field_address(int adjustment);

		// Attributes
		public:
			long *value;
};

////////////////////////////////////////////////////////////////////
// MoneyField class
////////////////////////////////////////////////////////////////////

class MoneyField : public Field
{
		// Constructor / Destructor
		public:
							MoneyField(Terminal *ioP,int x,int y,int
size,int64_t *value = NULL);
							MoneyField(Terminal *ioP,int size,int64_t
*value = NULL);
		virtual		~MoneyField();

		// Functions
		public:
			int reset();
			int backspace();
			int finalize_field();
			int add_char(int key);
			int display_field(int use_underscores = 0);
			int seen_dollar,seen_sign,seen_dot,seen_digit;

			void		update_field_address(int adjustment);

		// Attributes
		public:
			int64 *value;
};

////////////////////////////////////////////////////////////////////
// TextField class
////////////////////////////////////////////////////////////////////

class TextField : public Field
{
		// Constructor
		public:
			TextField(Terminal *ioP,int x,int y,int size,char *value = NULL);
			TextField(Terminal *ioP,int size,char *value = NULL);

		// Functions
		public:
			int add_char(int key);

			void		update_field_address(int adjustment);

		// Attributes
		public:
			char *value;
};

////////////////////////////////////////////////////////////////////
// Functions
////////////////////////////////////////////////////////////////////

Field *genfield(Terminal *ioP,int x,int y,int len,int *ptr);
Field *genfield(Terminal *ioP,int x,int y,int len,short *ptr);
Field *genfield(Terminal *ioP,int x,int y,int len,long *ptr);
Field *genfield(Terminal *ioP,int x,int y,int len,char *ptr);
Field *genfield(Terminal *ioP,int x,int y,int len,unsigned char *ptr);
Field *genfield(Terminal *ioP,int x,int y,int len,int64_t *ptr);

#endif
////////////////////////////////////////////////////////////////////
```

tuxClientInf.cpp

```
/*
****************************************************************
** Project                      : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                         : tuxClientInf.cpp
** Title                        : Abstraction layer for TUX client
****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                     :
**
****************************************************************
*/

#include <tuxClientInf.hpp>


FILE *logfile;

////////////////////////////////////////////////////////////////////
// Tuxedo interface
////////////////////////////////////////////////////////////////////

/*
****************************************************************
** Name                         : tuxBufferAllocate
** Description:
**                                                              Allocate buffer
** Parameters :
**                                                              thread_info_t * - thread
** Returns              :
**                                                              None
** Comments             :
**                                                              We calculate the max size
needed and allocate
**                                                              this amount of buffer space
**
****************************************************************
*/

void tuxBufferAllocate(thread_info_t * thread_context)
{
		int size = sizeof(dbInfo_data_t);

		(size < sizeof(newOrder_data_t))                ? sizeof(newOrder_data_t)
: size;
		(size < sizeof(payment_data_t))                 ? sizeof(payment_data_t)
: size;
		(size < sizeof(orderStatus_data_t))     ? sizeof(orderStatus_data_t)   : size;
		(size < sizeof(delivery_data_t))                ? sizeof(delivery_data_t)
: size;
		(size < sizeof(stockLevel_data_t))      ? sizeof(stockLevel_data_t)    : size;

		thread_context->tran_buffer_size        = size;
		thread_context->tran_buffer             =  tuxBufferAllocate(size);
		return;
}

/*
****************************************************************
** Name                         : tuxBufferAllocate
** Description:
**                                                              Allocate buffer
** Parameters :
**                                                              int - size
** Returns              :
**                                                              void * - buffer pointer
** Comments             :
**
****************************************************************
*/

void * tuxBufferAllocate(int size)
{
		void *buffer = NULL;

		if(null_db)
			buffer =  (void *) new char[size + 1];
		else buffer = (void *) tpalloc("CARRAY",NULL,size + 1);
```

```
        bzero(buffer,size+1); return buffer;
}

/*
****************************************************************
** Name                                : tuxBufferDeallocate
** Description:
**                                                              Deallocate buffer
** Parameters :
**                                                              thread_info_t * - thread
** Returns             :
**                                                              None
** Comments            :
**
****************************************************************
*/

void tuxBufferDeallocate(thread_info_t * thread_context)
{
        tuxBufferDeallocate(thread_context->tran_buffer);

        thread_context->tran_buffer_size        = 0;
        thread_context->tran_buffer             =  NULL;
        return;
}

/*
****************************************************************
** Name                                : tuxBufferDeallocate
** Description:
**                                                              Deallocate buffer
** Parameters :
**                                                              void * - buffer pointer
** Returns             :
**                                                              None
** Comments            :
**
****************************************************************
*/

void  tuxBufferDeallocate(void *buffer)
{
        if(null_db)
                delete buffer;

        tpfree((char *) buffer);
}

/*
****************************************************************
** Name                                : tuxClientConnect
** Description:
**                                                              Connect to TUX server
** Parameters :
**                                                              int - user
** Returns             :
**                                                              int - Return code
** Comments            :
**
****************************************************************
*/

int tuxClientConnect(int user_id)
{
        if(!null_db)
        {
                if(tpinit((TPINIT *) NULL) == ERR)
                        return ERR;
        }

        return OK;
}

/*
****************************************************************
** Name                                : tuxClientDisconnect
** Description:
**                                                              Disconnect from a TUX server
** Parameters :
**                                                              int user id
** Returns             :
**                                                              int - Return code
** Comments            :
**
****************************************************************
*/

int tuxClientDisconnect(int user_id)
{
        if(!null_db)
                tpterm();

        return OK;
}

/*
****************************************************************
** Name                                : tux_db_info
** Description:
**                                                              Client interface to get Database
Info
** Parameters :
**                                                              dbInfo_data_t * - transaction
data
**                                                              trpc_status_t * - status
** Returns             :
**                                                              dbInfo_data_t * - result buffer
** Comments            :
**
****************************************************************
*/

dbInfo_data_t * tux_db_info(dbInfo_data_t * dataP,trpc_status_t * trpcStatus)
{
        if (null_db)
        {
                tux_db_info_null_db(dataP);
                *trpcStatus = 0; return dataP;
        }
        else
        {
                long bufferSize = sizeof(dbInfo_data_t);

                int rc = tpcall("INFO",(char *) dataP,bufferSize,(char **)
&dataP,&bufferSize,0);
                #if CHECK_RC
                        if ( rc != OK )
                        {
                                fprintf(stderr,"TUXEDO tpcall info error:%d
transtatus:%d\n",tperrno,dataP->header.sql_code);
                        }
                #endif
                *trpcStatus = rc;
        }

        if(dataP->header.sql_code == OK && *trpcStatus != OK)
                dataP->header.sql_code = RPC_ERROR;

        return dataP;
}

/*
****************************************************************
** Name                                : tux_new_order
** Description:
**                                                              Client interface to do new order
transaction
** Parameters :
**                                                              thread_info_t * - context
**                                                              newOrder_data_t * - transaction
data
**                                                              trpc_status_t * - status
** Returns             :
**                                                              newOrder_data_t * - result
buffer
** Comments            :
**
****************************************************************
*/

newOrder_data_t * tux_new_order(thread_info_t * thread_context,newOrder_data_t * dataP,trpc_status_t *
trpcStatus)
{
        if (null_db)
        {
                tux_new_order_null_db(dataP);
                *trpcStatus = 0; return dataP;
        }
        else
        {
                int rc = tpcall("ORDER",(char *) dataP,sizeof(newOrder_data_t),

(char **) &thread_context->tran_buffer,&thread_context->tran_buffer_size,0);

                memcpy(dataP,thread_context->tran_buffer,sizeof(newOrder_data_t));
                #if CHECK_RC
                        if ( rc != OK )
                        {
                                fprintf(stderr,"TUXEDO tpcall order error:%d
transtatus%d\n",tperrno,dataP->header.sql_code);
                        }
                #endif
                *trpcStatus = rc;
        }

        if(dataP->header.sql_code == OK && *trpcStatus != OK)
        {
                dataP->header.sql_code = RPC_ERROR;
                fprintf(stderr,"New Order tpcall
error,s_transtatus:%d\n",dataP->header.sql_code);
        }
        return dataP;
}

/*
****************************************************************
** Name                                : tux_payment
** Description:
```

```
**                                                              Client interface to do payment
transaction
** Parameters :
**                                                              thread_info_t * - context
**                                                              payment_data_t * - transaction
data
**                                                              trpc_status_t * - status
** Returns                      :
**                                                              payment_data_t * - result buffer
** Comments                     :
**
*****************************************************************
*/

payment_data_t * tux_payment(thread_info_t * thread_context,payment_data_t * dataP,trpc_status_t *
trpcStatus)
{
        if (null_db)
        {
                tux_payment_null_db(dataP);
                *trpcStatus = 0; return dataP;
        }
        else
        {
                int rc = tpcall("PAYMENT",(char *) dataP,sizeof(payment_data_t),

(char **) &thread_context->tran_buffer,&thread_context->tran_buffer_size,0);

                memcpy(dataP,thread_context->tran_buffer,sizeof(payment_data_t));
                #if CHECK_RC
                        if ( rc != OK )
                        {
                                fprintf(stderr,"TUXEDO tpcall payment
error:%d  transtatus%d\n",tperrno,dataP->header.sql_code);
                        }
                #endif
                *trpcStatus = rc;
        }

        if(dataP->header.sql_code == OK && *trpcStatus != OK)
        {
                dataP->header.sql_code = RPC_ERROR;
                fprintf(stderr,"Payment tpcall
error,s_transtatus:%d\n",dataP->header.sql_code);
        }
        return dataP;
}
/*
*****************************************************************
** Name                                 : tux_order_status
** Description:
**                                                              Client interface to do order
status transaction
** Parameters :
**                                                              thread_info_t * - context
**                                                              orderStatus_data_t * -
transaction data
**                                                              trpc_status_t * - status
** Returns                      :
**                                                              orderStatus_data_t * - result
buffer
** Comments                     :
**
*****************************************************************
*/

orderStatus_data_t * tux_order_status(thread_info_t * thread_context,orderStatus_data_t *
dataP,trpc_status_t * trpcStatus)
{
        if (null_db)
        {
                tux_order_status_null_db(dataP);
                *trpcStatus = 0; return dataP;
        }
        else
        {
                int rc = tpcall("STATUS",(char *) dataP,sizeof(orderStatus_data_t),

(char **) &thread_context->tran_buffer,&thread_context->tran_buffer_size,0);

                memcpy(dataP,thread_context->tran_buffer,sizeof(orderStatus_data_t));
                #if CHECK_RC
                        if ( rc != OK )
                        {
                                fprintf(stderr,"TUXEDO tpcall order status
error:%d  transtatus%d\n",tperrno,dataP->header.sql_code);
                        }
                #endif
                *trpcStatus = rc;
        }

        if(dataP->header.sql_code == OK && *trpcStatus != OK)
        {
                dataP->header.sql_code = RPC_ERROR;
                fprintf(stderr,"Order Status tpcall
error,s_transtatus:%d\n",dataP->header.sql_code);
        }
        return dataP;
}

/*
*****************************************************************
** Name                                 : tux_delivery
** Description:
**                                                              Client interface to do delivery
transaction
** Parameters :
**                                                              thread_info_t * - context
**                                                              delivery_data_t * - transaction
data
**                                                              trpc_status_t * - status
** Returns                      :
**                                                              delivery_data_t * - result buffer
** Comments                     :
**
*****************************************************************
*/

delivery_data_t * tux_delivery(thread_info_t * thread_context,delivery_data_t * dataP,trpc_status_t *
trpcStatus)
{
        struct timeval  start_process;
        struct timezone         tp;

        if (null_db)
        {
                tux_delivery_null_db(dataP);
                *trpcStatus = 0; return dataP;
        }
        else
        {
                gettimeofday(&start_process,&tp);
                dataP->header.start_time.sec  =         start_process.tv_sec;
                dataP->header.start_time.usec           =
start_process.tv_usec;

                int rc = tpacall("DELIVERY",(char *)
dataP,sizeof(delivery_data_t),TPNOREPLY);
                #if CHECK_RC
                        if ( rc != OK )
                        {
                                fprintf(stderr,"TUXEDO tpacall delivery
error:%d transtatus%d\n",tperrno,dataP->header.sql_code);
                        }
                #endif
                *trpcStatus = rc;

        }

        if(dataP->header.sql_code == OK && *trpcStatus != OK)
        {
                dataP->header.sql_code = RPC_ERROR;
                fprintf(stderr,"Delivery tpacall
error,s_transtatus:%d\n",dataP->header.sql_code);
        }

        return dataP;
}

/*
*****************************************************************
** Name                                 : tux_stock_level
** Description:
**                                                              Client interface to do stock
level transaction
** Parameters :
**                                                              thread_info_t * - context
**                                                              stockLevel_data_t * -
transaction data
**                                                              trpc_status_t * - status
** Returns                      :
**                                                              stockLevel_data_t * result
buffer
** Comments                     :
**
*****************************************************************
*/

stockLevel_data_t * tux_stock_level(thread_info_t * thread_context,stockLevel_data_t *
dataP,trpc_status_t * trpcStatus)
{
        if (null_db)
        {
                tux_stock_level_null_db(dataP);
                *trpcStatus = 0; return dataP;
        }
        else
        {
                int rc = tpcall("STOCK",(char *) dataP,sizeof(stockLevel_data_t),

(char **) &thread_context->tran_buffer,&thread_context->tran_buffer_size,0);

                memcpy(dataP,thread_context->tran_buffer,sizeof(stockLevel_data_t));
                #if CHECK_RC
                        if ( rc != OK )
```

```
                                        {
                                                fprintf(stderr,"TUXEDO tpcall stock error:%d
transtatus%d\n",tperrno,dataP->header.sql_code);
                                        }
                        #endif
                        *trpcStatus = rc;
        }

        if(dataP->header.sql_code == OK && *trpcStatus != OK)
        {
                        dataP->header.sql_code = RPC_ERROR;
                        fprintf(stderr,"Stock tpcall error,s_transtatus:%d\n",dataP->header.sql_code);
        }
        return dataP;
}

//////////////////////////////////////////////////////////////////
```

tuxClientInf.hpp

```
/*
********************************************************************
** Project                      : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                                 : tuxClientInf.hpp
** Title                                : Abstraction layer for TUX client
********************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                             :
**
********************************************************************
*/

#ifndef __tuxClientInf_hpp__
#define __tuxClientInf_hpp__

#include <commonTuxDB2.hpp>
#include <tuxClientNullDB.hpp>

//////////////////////////////////////////////////////////////////
// Definitions
//////////////////////////////////////////////////////////////////

#define NUM_STATES
thread_state_done

#define TUXEDO_MAX_STATUS_STRING_SIZE        256

//////////////////////////////////////////////////////////////////
// Enumerations
//////////////////////////////////////////////////////////////////

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

typedef enum
{
        TPCC_SUCCESS
                                        = 0,
        GOOD_INPUT
                                                = 0,

        INVALID_NEWO
                                        = 100,
        SQL_ERROR
                                        = 2,
        TRPC_ERROR
                                                = 3,
        TUX_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;

////////////////////////////////////////////////////////////////
// Client structures
////////////////////////////////////////////////////////////////

typedef struct
{
  int                                                   server;
  int                                                   terminal;
  int                                                   tran;
  int                                                   sub_tran;

  struct timeval        start;
  struct timeval        send;
  struct timeval        srvr_start;
  struct timeval        srvr_done;
  struct timeval        end;

  int                                                   num_rms;
  int                                                   tran_failed;
} tran_timing_t;

typedef struct
{
  thread_state_t        state;

  int                                                   tran;

  struct timeval                init;
        struct timeval          called;
        struct timeval          sent;
        struct timeval          received;
        struct timeval          returned;
        struct timeval          done;

  int                                                   printed;
        int                                             done_printed;
} thread_descr_t;

typedef struct
{
   int                                                  num;
   int                                                  errs;

   double                                       RTtotal[2];

   int                                                  RTcount;
} tran_info_t;

typedef struct
{
   tran_info_t          tran[MAX_TRAN_TYPE + 1];

   int                                                  errors;

   double                                       time;
} total_tran_count_t;

typedef struct
{
   int                                                  status;
   int                                                  line;

   char                                                 file[268];

   unsigned long                tuxedoError;

   char
errorMsg[TUXEDO_MAX_STATUS_STRING_SIZE];
} tux_status_t;

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];

        void *                                          tran_buffer;
        long                                                    tran_buffer_size;

   int                                                  done;
} thread_info_t;

////////////////////////////////////////////////////////////////
// Function Definitions
////////////////////////////////////////////////////////////////

void * tuxBufferAllocate(int size);
void tuxBufferDeallocate(void *buffer);

void tuxBufferAllocate(thread_info_t * thread_contex);
void tuxBufferDeallocate(thread_info_t * thread_contex);

int tuxClientConnect(int user_id);
int tuxClientDisconnect(int user_id);
dbInfo_data_t * tux_db_info(dbInfo_data_t * dataP,trpc_status_t * trpcStatus);

// TPCC transactions
dbInfo_data_t * tux_db_info(dbInfo_data_t * dataP,trpc_status_t * trpcStatus);
newOrder_data_t * tux_new_order(thread_info_t * thread_context,newOrder_data_t * dataP,trpc_status_t *
trpcStatus);
payment_data_t * tux_payment(thread_info_t * thread_context,payment_data_t * dataP,trpc_status_t *
trpcStatus);
orderStatus_data_t * tux_order_status(thread_info_t * thread_context,orderStatus_data_t *
dataP,trpc_status_t * trpcStatus);
delivery_data_t * tux_delivery(thread_info_t * thread_context,delivery_data_t * dataP,trpc_status_t *
trpcStatus);
stockLevel_data_t * tux_stock_level(thread_info_t * thread_context,stockLevel_data_t *
dataP,trpc_status_t * trpcStatus);

#endif
////////////////////////////////////////////////////////////////
```

tuxClientInfUtilities.cpp

```
/*
****************************************************************
** Project                      : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                                 : tuxClientInfUtilities.cpp
** Title                                : Tux client interface helpers
****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                             :
**
****************************************************************
*/

#include <tuxClientInfUtilities.hpp>

/*
****************************************************************
** Name                                 : client_tpcc_rc_to_str
** Description:
**                                                              Translate return code to string
** Parameters :
**                                                              tpcc_rc_t - return code
** Returns                      :
**                                                              char * - Return code string
** Comments                     :
**                                                              Looks a little unfinished.
** Notes        :
**
****************************************************************
*/

char *client_tpcc_rc_to_str(tpcc_rc_t rc)
{
        static char retVal[10] = {"-Unknown-"};
```

```
        switch (rc)
        {
                case INVALID_NEWO:
                                                        return retVal;
                case INVALID_HANDLE:
                                                        return retVal;
                case SQL_ERROR:
                                                                return retVal;
                case TUX_ERROR:
                                                                return retVal;
                case TRPC_ERROR:
                                                                return retVal;
                case NO_SUCH_LAST_NAME:
                                                        return retVal;
                case INVALID_TRAN_TYPE:
                                                        return retVal;
                case TPCC_ERROR_BEGIN_NEWO:
                                return retVal;
                case TPCC_ERROR_DECL_NEWO_SEL_ITEM:
                return retVal;
                case TPCC_ERROR_OPEN_NEWO_SEL_ITEM:
                return retVal;
                case TPCC_ERROR_OPEN_DIST_NEWO_SEL_ITEM:
return retVal;
                case TPCC_ERROR_FETCH_NEWO_SEL_ITEM:
        return retVal;
                case TPCC_ERROR_FETCH_DIST_NEWO_SEL_ITEM:
return retVal;
                case TPCC_ERROR_PREP_NEWO_SEL_STCK:
                return retVal;
                case TPCC_ERROR_DECL_NEWO_SEL_STCK:
                return retVal;
                case TPCC_ERROR_OPEN_NEWO_SEL_STCK:
                return retVal;
                case TPCC_ERROR_OPEN_DIST_NEWO_SEL_STCK:
return retVal;
                case TPCC_ERROR_FETCH_NEWO_SEL_STCK:
        return retVal;
                case TPCC_ERROR_FETCH_DIST_NEWO_SEL_STCK:
return retVal;
                case TPCC_ERROR_NEWO_SELECT:
                                return retVal;
                case TPCC_ERROR_NEWO_UPD_STCK:
                return retVal;
                case TPCC_ERROR_DIST_NEWO_UPD_STCK:
                return retVal;
                case TPCC_ERROR_NEWO_SELECT_2:
                return retVal;
                case TPCC_ERROR_DECL_NEWO_SEL_CUST:
                return retVal;
                case TPCC_ERROR_OPEN_NEWO_SEL_CUST:
                return retVal;
                case TPCC_ERROR_OPEN_DIST_NEWO_SEL_CUST:
return retVal;
                case TPCC_ERROR_FETCH_NEWO_SEL_CUST:
        return retVal;
                case TPCC_ERROR_FETCH_DIST_NEWO_SEL_CUST:
return retVal;
                case TPCC_ERROR_DECL_NEWO_SEL_DIST:
                return retVal;
                case TPCC_ERROR_OPEN_NEWO_SEL_DIST:
                return retVal;
                case TPCC_ERROR_OPEN_DIST_NEWO_SEL_DIST:
return retVal;
                case TPCC_ERROR_FETCH_NEWO_SEL_DIST:
        return retVal;
                case TPCC_ERROR_FETCH_DIST_NEWO_SEL_DIST:
return retVal;
                case TPCC_ERROR_PREP_NEWO_INS_OL:
                return retVal;
                case TPCC_ERROR_DECL_NEWO_INS_OL:
                return retVal;
                case TPCC_ERROR_OPEN_NEWO_INS_OL:
                return retVal;
                case TPCC_ERROR_OPEN_DIST_NEWO_INS_OL:
        return retVal;
                case TPCC_ERROR_PUT_NEWO_INS_OL:
                                return retVal;
                case TPCC_ERROR_PUT_DIST_NEWO_INS_OL:
        return retVal;
                case TPCC_ERROR_DECL_NEWO_SEL_WARE:
                return retVal;
                case TPCC_ERROR_OPEN_NEWO_SEL_WARE:
                return retVal;
                case TPCC_ERROR_OPEN_DIST_NEWO_SEL_WARE:
return retVal;
                case TPCC_ERROR_FETCH_NEWO_SEL_WARE:
        return retVal;
                case TPCC_ERROR_FETCH_DIST_NEWO_SEL_WARE:
        return retVal;
                case TPCC_ERROR_EXECUTE_NEWO_UPD_INS:
        return retVal;
                case TPCC_ERROR_UPDATE_NEWO_NEXT_OID:
        return retVal;
                case TPCC_ERROR_PREP_NEWO_INS:
                return retVal;
                case TPCC_ERROR_EXECUTE_DIST_NEWO_INS:
        return retVal;
                case TPCC_ERROR_EXECUTE_NEWO_COMMIT:
        return retVal;
                case TPCC_ERROR_ROLLBACK_NEWO:
                return retVal;
                case TPCC_ERROR_REMOTE_OL_SELECT:
                return retVal;
                case TPCC_ERROR_REMOTE_OL_UPDATE:
                return retVal;
                case TPCC_ERROR_OPEN_ORDS_CNT_CID:
                return retVal;
                case TPCC_ERROR_FETCH_ORDS_CNT_CID:
                return retVal;
                case TPCC_ERROR_OPEN_ORDS_SEL_CLAST:
        return retVal;
                case TPCC_ERROR_FETCH_ORDS_SEL_CLAST:
        return retVal;
                case TPCC_ERROR_OPEN_ORDS_SEL_CID:
                return retVal;
                case TPCC_ERROR_FETCH_ORDS_SEL_CID:
                return retVal;
                case TPCC_ERROR_OPEN_ORDS_SEL_OLDORD:
        return retVal;
                case TPCC_ERROR_FETCH_ORDS_OLDORD:
                return retVal;
                case TPCC_ERROR_OPEN_ORDS_SEL_OL:
                return retVal;
                case TPCC_ERROR_FETCH_ORDS_SEL_OL:
                return retVal;
                case TPCC_ERROR_EXECUTE_ORDS_COMMIT:
        return retVal;
                case TPCC_ERROR_OPEN_DELIVERY_OLDEST_OID:
return retVal;
                case TPCC_ERROR_FETCH_DELIVERY_OLDEST_OID:
return retVal;
                case TPCC_ERROR_EXECUTE_DELIVERY_COMMIT:
return retVal;
                case TPCC_ERROR_OPEN_DELIVERY_SEL_ORD:
        return retVal;
                case TPCC_ERROR_FETCH_DELIVERY_SEL_ORD:
return retVal;
                case TPCC_ERROR_OPEN_DELIVERY_SEL_SUM_OL:
return retVal;
                case TPCC_ERROR_FETCH_DELIVERY_SEL_SUM_OL:
return retVal;
                case TPCC_ERROR_EXECUTE_DELIVERY_EXEC_DVRY:
return retVal;
                case TPCC_ERROR_SELECT_DELIVERY_ORDER_ID:
return retVal;
                case TPCC_ERROR_SELECT_DELIVERY_CARRIER_ID:
return retVal;
                case TPCC_ERROR_SELECT_DELIVERY_BALANCE:
return retVal;
                case TPCC_ERROR_OPEN_STOCKLEVEL_SEL_OID:
return retVal;
                case TPCC_ERROR_FETCH_STOCKLEVEL_SEL_OID:
return retVal;
                case TPCC_ERROR_OPEN_STOCKLEVEL_CNT_SID:
return retVal;
                case TPCC_ERROR_FETCH_STOCKLEVEL_CNT_SID:
return retVal;
                case TPCC_ERROR_OPEN_STOCKLEVEL_FIND:
        return retVal;
                case TPCC_ERROR_FETCH_STOCKLEVEL_FIND:
        return retVal;
                case TPCC_ERROR_EXECUTE_STOCKLEVEL_COMMIT:
return retVal;
                case TPCC_ERROR_OPEN_PAYMENT_CNT_CID:
        return retVal;
                case TPCC_ERROR_FETCH_PAYMENT_CNT_CID:
        return retVal;
                case TPCC_ERROR_OPEN_PAYMENT_SEL_CLAST:
return retVal;
                case TPCC_ERROR_FETCH_PAYMENT_SEL_CLAST:
return retVal;
                case TPCC_ERROR_OPEN_PAYMENT_SEL_CID:
        return retVal;
                case TPCC_ERROR_FETCH_PAYMENT_SEL_CID:
        return retVal;
                case TPCC_ERROR_DECL_PAYMENT_SEL_DIST:
        return retVal;
                case TPCC_ERROR_OPEN_PAYMENT_SEL_DIST:
        return retVal;
                case TPCC_ERROR_OPEN_DIST_PAYMENT_SEL_DIST:
return retVal;
                case TPCC_ERROR_FETCH_PAYMENT_SEL_DIST:
return retVal;
                case TPCC_ERROR_FETCH_DIST_PAYMENT_SEL_DIST:
return retVal;
                case TPCC_ERROR_DECL_PAYMENT_SEL_WARE:
        return retVal;
                case TPCC_ERROR_OPEN_PAYMENT_SEL_WARE:
        return retVal;
                case TPCC_ERROR_OPEN_DIST_PAYMENT_SEL_WARE:
return retVal;
                case TPCC_ERROR_FETCH_PAYMENT_SEL_WARE:
return retVal;
                case TPCC_ERROR_FETCH_DIST_PAYMENT_SEL_WARE:
return retVal;
                case TPCC_ERROR_EXECUTE_PAYMENT_UPD_CUST_LAST:
return retVal;
```

```
                        case TPCC_ERROR_EXECUTE_PAYMENT_UPD_CUST_ID:
return retVal;
                        case TPCC_ERROR_COMMIT_PAYMENT_UPD_CUST:
                return retVal;
                        case TPCC_ERROR_SELECT_PAYMENT_W_YTD:
                return retVal;
                        case TPCC_ERROR_SELECT_PAYMENT_D_YTD:
                return retVal;
                        case TPCC_ERROR_BEGIN_PAYMENT:
                        return retVal;
                        case TPCC_ERROR_EXECUTE_PAYMENT_COMMIT:
return retVal;
                        default:

return retVal;
                }

                return NULL;
}

/*
****************************************************************
** Name                                 : client_tran_code_to_name
** Description:
**                                                              Translate transaction code to
name
** Parameters :
**                                                              int - transaction code
** Returns              :
**                                                              char * - transaction name
** Comments             :
**
** Notes        :
**
****************************************************************
*/

char *client_tran_code_to_name(int type)
{
                static char retVal[6][10] =
                        {
                                "-Unknown-",
                                "NEWOR",
                                "PAYMN",
                                "ORDER",
                                "DELIV",
                                "STOCK"
                        };

                switch(type)
                {
                        case NEW_ORDER_TRANS:                   return retVal[1];
                        case PAYMENT_TRANS:                     return retVal[2];
                        case ORDER_STATUS_TRANS:        return retVal[3];
                        case DELIVERY_TRANS:                    return retVal[4];
                        case STOCK_TRANS:
return retVal[5];
                }

                return retVal[0];
}

/*
****************************************************************
** Name                                 : client_report_return
** Description:
**                                                              Report return status
** Parameters :
**                                                              char * - message
**                                                              data_header * - status
** Returns              :
**                                                              None
** Comments             :
**
** Notes        :
**
****************************************************************
*/

void client_report_return(char *msg,data_header *statusP)
{
                char            logmsg[MAX_MSG_LEN];

                switch (statusP->returncode)
                {
                case SUCCESS_CODE:
                        sprintf(logmsg,"After %s, rc = %d\n",msg,statusP->returncode);
                        err_printf(logmsg);
                        break;
                case SQL_ERROR:
                        sprintf(logmsg,"ERROR: After %s, rc = SQL_ERROR, SQL=%d,
ISAM=%d\n",

msg,statusP->sql_code,statusP->isam_code);
                        err_printf(logmsg);
                        break;
                case INVALID_NEWO:
                        sprintf(logmsg,"After %s, rc = INVALID_NEWO\n",msg);
                        err_printf(logmsg);
                        break;
                case TUX_ERROR:
                        sprintf(logmsg,"ERROR: After %s, rc = TUX_ERROR\n",msg);
                        err_printf(logmsg);
                        break;
                case TRPC_ERROR:
                        sprintf(logmsg,"ERROR: After %s, rc = TRPC_ERROR\n",msg);
                        err_printf(logmsg);
                        break;
                case NO_SUCH_LAST_NAME:
                        sprintf(logmsg,"After %s, rc = NO_SUCH_LAST_NAME.\n",msg);
                        err_printf(logmsg);
                        break;
                case DISTRIBUTED_TRAN_FAILED:
                        sprintf(logmsg,"After %s, rc = DISTRIBUTED_TRAN_FAILED.\n",msg);
                        err_printf(logmsg);
                        break;
                default:
                        sprintf(logmsg,"ERROR: After %s, rc = %s (%d), SQL=%d, ISAM=%d\n",

msg,client_tpcc_rc_to_str((tpcc_rc_t) statusP->returncode),statusP->returncode,

statusP->sql_code,statusP->isam_code);
                        err_printf(logmsg);
                        break;
                }

                return;
}

/*
****************************************************************
** Name                                 : client_set_return_code
** Description:
**                                                                      Set return status
** Parameters :
**                                                                      tpcc_data_t * - transaction
structure
**                                                                      tpcc_rc_t - result code;
** Returns              :
**                                                                      None
** Comments             :
**
** Notes        :
**
****************************************************************
*/

void client_set_return_code(tpcc_data_t *dataP,tpcc_rc_t code)
{
                switch (dataP->tran_type)
                {
                case NEW_ORDER_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_STATUS_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;
                }
}

/*
****************************************************************
** Name                                 : tux_error_message
** Description:
**                                                                      Log TUX error
** Parameters :
**                                                                      char * - error message
**                                                                      unsigned long - error code
** Returns              :
**                                                                      None
** Comments             :
**
** Notes        :
**
```

```
*****************************************************************
*/

void tux_error_message(char *msg,unsigned long err)
{
        char            logmsg[MAX_MSG_LEN];

        sprintf(logmsg,"ERROR: %s.  Error code = %d (0x%x)\n",msg,err,err);
        err_printf(logmsg);

        return;
}

/*
*****************************************************************
** Name                                    : tux_error
** Description:
**                                                         Log TUX error and exit
** Parameters :
**                                                         char * - function
**                                                         unsigned long - error code
** Returns                 :
**                                                         None
** Comments                :
**
** Notes      :
**
*****************************************************************
*/

void tux_error(char *function,unsigned long err)
{
        char error[TUXEDO_MAX_STATUS_STRING_SIZE];

        sprintf(error,"%s()",function);
        tux_error_message(error,err);

        exit_program(1); return;
}

/*
*****************************************************************
** Name       : tux_response
** Description :
**          Client interface to set transaction header
**          return codes
** Parameters  :
**          data_header - transaction data header
**          int - return code
**                                                         int             - sql_code
** Returns    :
**          None
** Comments    :
**
*****************************************************************
*/

void tux_response(data_header *header,int rc,int sql_code)
{
        //set return codes
        header->dtype                   =       0;
        header->returncode      =       rc;
        header->sql_code                =       sql_code;
        header->isam_code               =       0;
        header->num_rms                 =       0;
        header->stats                   =       0;

}

//////////////////////////////////////////////////////////////////////
```

tuxClientInfUtilities.hpp

```
/*
*****************************************************************
** Project                 : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                                    : tuxClientInfUtilities.hpp
** Title                                   : Tux client interface helpers
*****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**          Developed at IBM Austin by the AIX RS/6000
**          performance group.
**
** Comments                                :
**
*****************************************************************
*/

#ifndef __tuxClientInfUtilities_hpp__
#define __tuxClientInfUtilities_hpp__

#include <commonTuxDB2.hpp>
#include <tuxClientInf.hpp>
#include <tuxClientMonitor.hpp>

//////////////////////////////////////////////////////////////////////
// Definitions
//////////////////////////////////////////////////////////////////////

#define ENDMSG
                        "...."

//////////////////////////////////////////////////////////////////////
// Macros
//////////////////////////////////////////////////////////////////////

#define ERROR(x)

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

#define CHK_STATUS(status,val,a)
                                                                \
        if(status)

                                                \
        {

                                                                \
                exit_program(status);

        \
        }

//////////////////////////////////////////////////////////////////////
// Function Definitions
//////////////////////////////////////////////////////////////////////

void client_report_return(char *msg,data_header *statusP);
void client_set_return_code(tpcc_data_t *dataP,tpcc_rc_t code);

char *client_tran_code_to_name(int type);
char *client_tpcc_rc_to_str(tpcc_rc_t rc);

void tux_error_message(char *msg,unsigned long err);
void tux_error(char *function,unsigned long err);

#endif
//////////////////////////////////////////////////////////////////////
```

tuxClientListen.cpp

```
/*
*****************************************************************
** Project                 : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                                    : tuxClientListen.cpp
** Title                                   : Listner
*****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**          Developed at IBM Austin by the AIX RS/6000
**          performance group.
**
** Comments                                :
**
*****************************************************************
*/

#include <tuxClientListen.hpp>

/*
*****************************************************************
** Name                                    : process_terminal
** Description:
**                                                         Process terminal interaction
** Parameters :
**                                                         int - connection
** Returns                 :
**                                                         None
** Comments                :
**                                                         This used a try/catch block
which I don't think
**                                                         works in this situation. should
really be changed
**                                                         to check the return codes as the
system routines
```

```
**                                                              will not throw an exception.
**
*****************************************************************
*/

void process_terminal(int fdIn)
{
        pthread_t                                               thread
                = pthread_self();
        tid_t                                                                   tid
                                                                = thread_self();

        terminal_context_t      terminal_context;
        tpcc_data_t                                             tran_data;

        int
w_id;
        int
d_id;
        int
thread_id                                       = pthread_getunique_np(&thread,&thread_id);

        struct timespec                         rand_sleep;

        char
logmsg[MAX_MSG_LEN];
        //  Default terminal context
        //  This may be updated later by the terminal

        terminal_context.terminal_id = -1;
        terminal_context.profiling              = 0;

        // begin client processing
        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();

                        sprintf(logmsg,"Tid: %d (0x%x) w_id %d, d_id
%d\n",tid,tid,w_id,d_id);
                        err_printf(logmsg);

client_init(fdIn,fdIn,w_id,d_id,terminal_context.handle_contextP);

                        sprintf(logmsg,"Thread done - Tid %d (0x%x)\n",tid,tid);
                        err_printf(logmsg);
                        clnt_thread_done(terminal_context.handle_contextP);
                }
                else
                {
                        sprintf(logmsg,"Starting Auxiliary Thread, Tid %d
(0x%x)\n",tid,tid);
                        err_printf(logmsg);

                        client_status_report(fdIn);
                        sprintf(logmsg,"End of Auxiliary Thread, Tid %d (0x%x)\n",
tid, tid);
                        err_printf(logmsg);
                }
        }
        catch(...)
        {
                sprintf(logmsg,"An exception happened\n");
                err_printf(logmsg);

                sprintf(logmsg,"End of Auxiliary Thread, Tid %d (0x%x)\n",tid,tid);
                logprintf(logmsg);
        }

        // finish up
        close(fdIn);
        return;
}

/*
*****************************************************************
** Name                                 : terminal_thread_entry
** Description:
**                                                                 Entry point for terminal thread
** Parameters :
**                                                                 void * - thread entry parameter
** Returns              :
**                                                                 void * - thread exit parameter
** Comments             :
**                                                                 Thread entry parameter will be
a file descriptor
**                                                                 and the memory for this value
will be released.
**
*****************************************************************
*/

void * terminal_thread_entry(void * threadParameter)
{
        int             *fileDescriptor         = (int *) threadParameter;

        // call terminal processing routines
        process_terminal(*fileDescriptor);

        // release file descriptor memory
        delete fileDescriptor;

        // thread terminated
        return NULL;
}

/*
*****************************************************************
** Name                                 : create_terminal_thread
** Description:
**                                                                 Create a terminal thread
** Parameters :
**                                                                 int - fdIn
** Returns              :
**                                                                 None
** Comments             :
**
*****************************************************************
*/

void create_terminal_thread(int fdIn)
{
        int                                                     *fileDescriptor;
        pthread_t               threadId;

        pthread_attr_t threadAttr;

        // create thread parameter which is safe to pass into create thread.
        // The thread entry point will release this memory.

        fileDescriptor = new int;
        *fileDescriptor = fdIn;

        //initialize thread attributes
        pthread_attr_init(&threadAttr);
        pthread_attr_setdetachstate(&threadAttr,PTHREAD_CREATE_UNDETACHED);

        //create the thread with start function and parameters
        pthread_create(&threadId,&threadAttr,terminal_thread_entry,(void *)fileDescriptor);

        //set flag to destroy thread's attribute object
        pthread_attr_destroy(&threadAttr);

        //set the thread to be a detached process
        pthread_detach(threadId);

        return;
}

/*
*****************************************************************
** Name                                 : set_terminal_socket_options
** Description:
**                                                                 Set the socket options we want
for the terminal
** Parameters :
**                                                                 int - fdIn
** Returns              :
**                                                                 None
** Comments             :
**
*****************************************************************
*/

void set_terminal_socket_options(int fdIn)
{
        int             tcpNoDelay                              = 1;
        int             reuseAddr                               = 1;

        setsockopt(fdIn,IPPROTO_TCP,TCP_NODELAY,(char *)
&tcpNoDelay,sizeof(tcpNoDelay));
        setsockopt(fdIn,SOL_SOCKET,SO_REUSEADDR,(char *) &reuseAddr,sizeof(reuseAddr));

        return;
}

/*
*****************************************************************
** Name                                 : make_connections
** Description:
**                                                                 Listen for new connections on a
socket
** Parameters :
**                                                                 int - port
** Returns              :
**                                                                 None
** Comments             :
**                                                                 Whenever a connection is
made, start a thread to talk
**                                                                 to the terminal.
**
**                                                                 There is currently no logic to
tell this function
**                                                                 to stop accepting more requests
```

```
and shutdown
**
*****************************************************************
*/

void make_connections(int port)
{
        int             rc;
        char            logmsg[MAX_MSG_LEN];

        // Debug/Status messages
        sprintf(logmsg,"Using socket %d\nUsing thread stack size default\n",port);
        err_printf(logmsg);

        // set up socket address
        sockaddr_in     socketAddress;
        size_t                  sockAddrLen = sizeof(sockaddr_in);

        bzero((void *) &socketAddress,sockAddrLen);

        socketAddress.sin_family                = AF_INET;
        socketAddress.sin_len                   = sockAddrLen;
        socketAddress.sin_addr.s_addr           = htonl(INADDR_ANY);
        socketAddress.sin_port                  = htons(port);

        // create the listner socket
        int                                     acceptSocket;

        acceptSocket = socket(AF_INET,SOCK_STREAM,0);
        if(!acceptSocket)
        {
                perror("Error socket() : ");
                exit_program(1);
        }

        // bind socket
        rc = bind(acceptSocket,(sockaddr *) &socketAddress,sockAddrLen);
        if(rc == ERR)
        {
                perror("Error bind() : ");
                exit_program(1);
        }

        rc = getsockname(acceptSocket,(sockaddr *) &socketAddress,&sockAddrLen);
        if(rc == ERR)
        {
                perror("Error getsockname() : ");
                exit_program(1);
        }

        sprintf(logmsg,"Connected and Bound to port %d, with socket %d\n",port,acceptSocket);
        err_printf(logmsg);

        rc = listen(acceptSocket,SOMAXCONN);
        if(rc == ERR)
        {
                perror("Error listen() : ");
                exit_program(1);
        }

        // accept connections
        sprintf(logmsg,"Accepting connection requests on port %d using socket
%d\n",port,acceptSocket);
        err_printf(logmsg);

        int terminalSocket;
        while(TRUE)
        {
                do
                {
                        terminalSocket = accept(acceptSocket,(sockaddr *)
&socketAddress,&sockAddrLen);
                } while(terminalSocket == ERR);

                // set terminal socket options
                set_terminal_socket_options(terminalSocket);

                // assign thread to handle terminal
                sprintf(logmsg,"Terminal conected to socket %d\n",terminalSocket);
                err_printf(logmsg);
                create_terminal_thread(terminalSocket);
        }

        // start listing for new connection requests
        sprintf(logmsg,"Make connections finished\n");
        err_printf(logmsg);
        return;
}
/*
*****************************************************************
** Name                                 : client_login
** Description:
**                                                              Login for client
** Parameters :
**                                                              int - file descriptor in
**                                                              int - file descriptor out
**                                                              int * - wherehouse
**                                                              int * - district
** Returns                      :
**                                                              int - status
** Comments                     :
**
*****************************************************************
*/

int client_login(int infd,int outfd,int *w_idP,int *d_idP)
{
        // Thread
        Thread_data thread(infd, outfd, NULL);

        // User
        User_data user_data;

        // login
        Login log(&user_data, &thread);
        log.handle();

        // set wherehouse and district
        *w_idP = user_data.warehouse;
        *d_idP = user_data.district;

        return OK;
}

/*
*****************************************************************
** Name                                 : client_init
** Description:
**                                                              Initlize client
** Parameters :
**                                                              int - file descriptor in
**                                                              int - file descriptor out
**                                                              int - wherehouse
**                                                              int - district
**                                                              void * - context
** Returns                      :
**                                                              int - status
** Comments                     :
**
*****************************************************************
*/

int client_init (int infd,int outfd,int w_id,int d_id,void *contextP)
{
        int rc = 0;

        // Create client
        Thread_data     *threadP                = new Thread_data(infd,outfd,contextP);
        Field                                   *menuField      = new IntField(threadP,8);

        User_data       user_data;
        Menu                                    menu(&user_data,threadP);

        // Initilize user data
        user_data.warehouse             = w_id;
        user_data.district              = d_id;
        menu.present();

        // Transaction classes
        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);

        // Process terminal interactions
        while(rc == OK)
        {
                int key = menuField->get_key();

                // process selection
                set_client_debug_state(contextP,1,key - '0');
                switch (key)
                {
                case EOF:
                        rc = ERR;
                        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(OK);
                                default:
                                        threadP->write("\a",1);
                                        break;
                                }
                                set_client_debug_state(contextP,4,key - '0');
                        }

                        return OK;
}

//////////////////////////////////////////////////////////////////////
```

tuxClientListen.hpp

```
/*
*********************************************************************
** Project                      : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                                 : tuxClientListen.hpp
** Title                                : Tux Client Listner
*********************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                             :
**
*********************************************************************
*/

#ifndef __tuxClientListen_hpp__
#define __tuxClientListen_hpp__

#include <commonTuxDB2.hpp>
#include <tuxClientCommon.hpp>
#include <tuxClientInf.hpp>
#include <tuxClientInfUtilities.hpp>

#include <tuxClientScreen.hpp>
#include <tuxClientField.hpp>
#include <tuxClientThread.hpp>

//////////////////////////////////////////////////////////////////////
// Structures
//////////////////////////////////////////////////////////////////////

typedef struct
{
                int             profiling;
                int             terminal_id;
                void            *handle_contextP;
} terminal_context_t;

//////////////////////////////////////////////////////////////////////
// Function Definitions
//////////////////////////////////////////////////////////////////////

void create_terminal_thread(int fdIn);
void process_terminal(int fdIn);

void create_terminal_thread(int fdIn);
void set_terminal_socket_options(int fdIn);
void make_connections(int port);

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

#endif
//////////////////////////////////////////////////////////////////////
```

tuxClientMonitor.cpp

```
/*
*********************************************************************
** Project                      : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                                 : tuxClientMonitor.cpp
** Title                                : Abstraction layer for TUX client
*********************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                             :
**
*********************************************************************
*/

#include <tuxClientMonitor.hpp>

/*
*********************************************************************
** Name                                 : pre_rpc
** Description:
**                                                                      Set thread state and track time
** Parameters :
**                                                                      thread_info_t * - thread info
**                                                                      data_header * - header
**                                                                      int - transaction type
**                                                                      int - sub transaction type
** Returns              :
**                                                                      None
** Comments             :
**                                                                      This is used by the Background
thread to track
**                                                                      the client stats.
**
*********************************************************************
*/

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;

                // no thread stucture
                if(!thread_infoP) return;

                // initilize thread timing structure
                curP = &thread_infoP->last_tran;

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

                // Check if we want stats
                if(iStatsFrequency >= 1)
                {
                        int num = ++(pClientInfo->tran[tran_type].num);
                        headerP->stats = (num % iStatsFrequency==0) ? 1 : 0;
                }
                else headerP->stats = 0;

                // Setup stucture to collect stats
                if(headerP->stats)
                {
                        gettimeofday(&curP->start, &tz);

                        headerP->start_time.sec = 0;
                        headerP->start_time.usec = 0;
                        headerP->end_time.sec = 0;
                        headerP->end_time.usec = 0;
                }
}

/*
*********************************************************************
** Name                                 : post_rpc
** Description:
**                                                                      Called after RPC to server
returns
** Parameters :
**                                                                      thread_info_t * - thread info
**                                                                      data_header * - header
**                                                                      int - transaction type
** Returns              :
**                                                                      None
** Comments             :
**                                                                      Record the client and server
response times and
**                                                                      tracks the client
** Notes                                :
**                                                                      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.
**
*********************************************************************
*/

void post_rpc(thread_info_t *thread_infoP,data_header *headerP,int tran_type)
{
                double                                                  time_diff_s;
                double                                                  time_diff_c;

                tran_timing_t           *curP;
                struct timezone         tz;
```

```
	// no thread stucture
	if(!thread_infoP) return;

	// current thread
	curP = &thread_infoP->last_tran;
	curP->server = headerP->dtype;

	// collecting time stats
	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);
	}

	// Store the info for each client.
	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 ++;
	}
}
/*
****************************************************************
** Name				: exit_program
** Description:
**					Exit
** Parameters :
**					short - error code
** Returns		:
**					None
** Comments		:
**					exit_program - restores original
terminal attributes
**					before leaving the program.
**
****************************************************************
*/

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

	// cancel all the longterm reservations (if any) and write out the time-stamps
	pthread_mutex_lock(&init_lock);

	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;
			}
		}
	}

	pthread_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);
		}
	}

	tuxClientDisconnect(user_id);
	exit(err); return;
}

/*
****************************************************************
** Name				: clnt_thread_init
** Description:
**					Initilizes each worker thread
** Parameters :
**					void * - thread info handle
** Returns		:
**					None
** Comments		:
**					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()
{
	int				thread_index;
	char
logmsg[MAX_MSG_LEN];

	struct timezone		tz;

	thread_info_t		*thread_infoP;

	// no statistics
	if(iStatsFrequency < 1)
		return(NULL);

	// allocate memory for thread info structure
	thread_infoP = (thread_info_t *) calloc(1,sizeof(thread_info_t));

	// initilize thread info structure
	thread_infoP->descr.state = thread_state_init;
	gettimeofday(&thread_infoP->descr.init,&tz);

	// get buffer to run TUX transactions
	tuxBufferAllocate(thread_infoP);
	thread_infoP->initialized	= 1;

	// add worker thread
	pthread_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;

	pthread_mutex_unlock(&init_lock);

	// thread create message every 200
	if(num_active_threads % 200 == 0)
	{
		sprintf(logmsg,"Thread %d Initialized (currently %d are active).\n",
				thread_index, num_active_threads);
		err_printf(logmsg);
	}

	// handle created
	return(thread_infoP);
}

/*
****************************************************************
** Name				: clnt_thread_done
** Description:
**					Exiting worker thread
```

```
** Parameters :
**                                                              void * - Worker handle
** Returns              :
**                                                              None
** Comments             :
**                                                              performs thread clean-up
**
*****************************************************************
*/

void clnt_thread_done(void *contextP)
{
        int                                                     infoListCount;
        int                                                     all_done = 0;

        thread_info_t **curP;
        thread_info_t  *infoP           = (thread_info_t *)contextP;

        if(!infoP) return;

        // release buffer to run TUX transactions
        tuxBufferDeallocate(infoP);

        // remove worker thread
        pthread_mutex_lock(&init_lock);

        set_client_debug_state((void *)infoP,thread_state_done,0);

        num_active_threads--;

        infoP->done = 1;

        // all worker threads gone?
        if(num_active_threads == 0)
        {
                all_done = 1;
        }

        // validity check
        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]);
        }


        // clean up list if all workers gone
        if(all_done)
        {
                for (infoListCount = 0,curP = info_list;infoListCount <
info_list_len;infoListCount++,curP++)
                        free(*curP);

                free(info_list);
                info_list = NULL;
                info_list_len = 0;
        }

        // release lock
        pthread_mutex_unlock(&init_lock);
        return;
}

/*
*****************************************************************
** Name                         : terminal_context_init
** Description:
**                                                              Initilizes the terminal
connection
** Parameters :
**                                                              int - Fikle descriptor?
** Returns              :
**                                                              void * - Worker handle
** Comments             :
**                                                              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;
        char
logmsg[MAX_MSG_LEN];
        struct timezone         tz;
        thread_info_t           *thread_infoP;

        // No statistics
        if(iStatsFrequency < 1)
                return(NULL);

        // allocate memory for thread info structure
        thread_infoP = (thread_info_t *) calloc(1,sizeof(thread_info_t));

        // initilize thread info structure
        thread_infoP->descr.state = thread_state_init;
        gettimeofday(&thread_infoP->descr.init, &tz);
        thread_infoP->initialized = 1;

        // take lock to add worker
        pthread_mutex_lock(&init_lock);

        // add worker thread
        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;

        // Unlock
        pthread_mutex_unlock(&init_lock);

        // thread create message every 200
        if(num_active_threads % 200 == 0)
        {
                sprintf(logmsg,"Terminal %d Initialized (currently %d are active).\n",
                                                        thread_index,num_active_threads);
                err_printf(logmsg);
        }

        // handle created
        return(thread_infoP);
}

/*
*****************************************************************
** Name                         : init_tux_client
** Description:
**                                                              Initilizes the tux client handle
** Parameters :
**                                                              int - user id
** Returns              :
**                                                              none
** Comments             :
**
*****************************************************************
*/

void init_tux_client(int user_id)
{
        int                                                             readSize;

        char                                                            *env_str;
        char                                                            serverName[48];
        char                                                            readBuffer[100];
        char
logmsg[MAX_MSG_LEN];

        unsigned long           status ;

        struct timezone         tz;
        struct timeval          a_time;

        FILE                                                            *rcFile;

        // allocate client memory
        if(pClientInfo == NULL)
                pClientInfo = (total_tran_count_t *) malloc(sizeof(total_tran_count_t));

        // initilize client memory
        memset(pClientInfo,0,sizeof(total_tran_count_t));

        // create lock mutex
        pthread_mutex_init(&init_lock,NULL);

        // initilize list
        info_list = NULL;
        info_list_len = 0;

        // initialize statistics
        iStatsFrequency = 1;

        // open stats file
        rcFile = fopen("~/.tpccrc","r");
        if(rcFile!=NULL)
        {
                // search file for "StatsFrequency"
                while (1)
                {
                        readSize = fscanf(rcFile,"%s",readBuffer);
                        if(readSize <= 0) break;
```

```
                                        if(strcasecmp(readBuffer,"StatsFrequency") == 0)
                                        {
                                                fscanf(rcFile,"%d",&iStatsFrequency);
                                                break;
                                        }
                                }
                        }

                        // display read "StatsFrequency"
                        sprintf(logmsg,"iStatsFrequency=%d\n",iStatsFrequency);
                        err_printf(logmsg);

                        // randomize
                        gettimeofday(&a_time,&tz);
                        srand48(a_time.tv_sec ^ a_time.tv_usec);

                        /// Initilize Tuxedo
                        if(tuxClientConnect(user_id) != OK)
                        {
                                char msg[100];
                                sprintf(msg,"TRPC error during TUX TPM init.");
                                printf("Unable to initilize TPM interface.\n");

                                exit_program(-1);
                        }

                        tuxClientConnect(user_id);
                        return;
}

/*
****************************************************************
** Name                                 : enroll_client
** Description:
**                                                              Obtains tp handles
** Parameters :
**                                                              int - user id
** Returns              :
**                                                              none
** Comments             :
**
****************************************************************
*/

void enroll_client(int user_id)
{
                        unsigned long  status ;
                        static  int             client_enrolled = 0;

                        // lock for initilization
                        pthread_mutex_lock(&init_lock);

                        // are we already enrolled?
                        if(client_enrolled)
                        {
                                // unlockand leave
                                pthread_mutex_unlock(&init_lock);
                                return;
                        }

                        // check we have a tp server and database

                        trpc_status_t trpcStatus;
                        dbInfo_data_t *data = (dbInfo_data_t *) tuxBufferAllocate(sizeof(dbInfo_data_t));

                        // request db info
                        data = tux_db_info(data,&trpcStatus);
                        tuxBufferDeallocate(data);

                        // did we get an error?
                        if(trpcStatus)
                        {
                                pthread_mutex_unlock(&init_lock);

                                char msg[100];
                                sprintf(msg, "TRPC error during db info at init.");
                                tux_error_message(msg,trpcStatus);

                                CHK_STATUS(33,NOINFO_TRPC_ERROR,"TRPC error during db info at
init");
                        }

                        client_enrolled = 1;
                        pthread_mutex_unlock(&init_lock);
}

/*
****************************************************************
** Name                                 : too_many_errors
** Description:
**                                                              Check number of errors
** Parameters :
**                                                              void * - contex
** Returns              :
**                                                              int - hard error
** Comments             :
**
****************************************************************
*/

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

  return (thread_context->consecutive_errors > MAX_CONSECUTIVE_ERRORS);
}

////////////////////////////////////////////////////////////////////
// Send functions called from the scree modle after the transaction
// data is recieved
////////////////////////////////////////////////////////////////////

/*
****************************************************************
** Name                                 : send_new_order
** Description:
**                                                                                 Send a new order request to the
server
** Parameters :
**                                                                                 void * - contex
**                                                                                 new_order_data_t * -
transaction data
** Returns              :
**                                                                                 new_order_data_t * - return
buffer
** Comments             :
**
****************************************************************
*/

newOrder_data_t * send_new_order(void * contextP,newOrder_data_t * dataP)
{
                        char                                                    logmsg[MAX_MSG_LEN];

                        trpc_status_t   trpcStatus;
                        thread_info_t  *threadContex                    = (thread_info_t *) contextP;

                        // output debug info
                        sprintf(logmsg,"New Order, w_id %d, %d
orders\n",dataP->in.s_W_ID,dataP->in.s_O_OL_CNT);
                        err_printf(logmsg);

                        // common pre transaction call
                        pre_rpc(threadContex,&dataP->header,NEW_ORDER_TRANS,(dataP->in.s_all_local ==
0));

                        // call new order client interface
                        dataP = tux_new_order(threadContex,dataP,&trpcStatus);

                        // common post transaction call
                        post_rpc(threadContex,&dataP->header,NEW_ORDER_TRANS);

                        return dataP;
}

/*
****************************************************************
** Name                                 : send_payment
** Description:
**                                                                                 Send a payment request to the
server
** Parameters :
**                                                                                 void * - contex
**                                                                                 payment_data_t * - transaction
data
** Returns              :
**                                                                                 payment_data_t * - return
buffer
** Comments             :
**
****************************************************************
*/

payment_data_t * send_payment(void * contextP,payment_data_t * dataP)
{
                        trpc_status_t   trpcStatus;
                        thread_info_t  *threadContex                    = (thread_info_t *) contextP;

                        // common pre transaction call
                        pre_rpc(threadContex,&dataP->header,PAYMENT_TRANS,(dataP->in.s_W_ID !=
dataP->in.s_C_W_ID));

                        // call payment client interface
                        dataP = tux_payment(threadContex,dataP,&trpcStatus);

                        // common post transaction call
                        post_rpc(threadContex,&dataP->header,PAYMENT_TRANS);

                        return dataP;
}

/*
****************************************************************
** Name                                 : send_order_status
** Description:
**                                                                                 Send a order status request to
the server
```

```
** Parameters :
**                                                      void * - contex
**                                                      orderStatus_data_t * -
transaction data
** Returns                      :
**                                                      orderStatus_data_t * - return
buffer
** Comments                     :
**
*****************************************************************
*/

orderStatus_data_t * send_order_status(void * contextP,orderStatus_data_t * dataP)
{
        trpc_status_t   trpcStatus;
        thread_info_t  *threadContex                    = (thread_info_t *) contextP;

        // common pre transaction call
        pre_rpc(threadContex,&dataP->header,ORDER_STATUS_TRANS,0);

        // call order status client interface
        dataP = tux_order_status(threadContex,dataP,&trpcStatus);

        // common post transaction call
        post_rpc(threadContex,&dataP->header,ORDER_STATUS_TRANS);

        return dataP;
}

/*
*****************************************************************
** Name                                 : send_delivery
** Description:
**                                                      Send a delivery request to the
server
** Parameters :
**                                                      void * - contex
**                                                      delivery_data_t * - transaction
data
** Returns                      :
**                                                      delivery_data_t * - return buffer
** Comments                     :
**
*****************************************************************
*/

delivery_data_t * send_delivery(void * contextP,delivery_data_t * dataP)
{
        trpc_status_t   trpcStatus;
        thread_info_t  *threadContex                    = (thread_info_t *) contextP;

        // common pre transaction call
        pre_rpc(threadContex,&dataP->header,DELIVERY_TRANS,0);

        // call delivery client interface
        dataP = tux_delivery(threadContex,dataP,&trpcStatus);

        // common post transaction call
        post_rpc(threadContex,&dataP->header,DELIVERY_TRANS);

        return dataP;
}

/*
*****************************************************************
** Name                                 : send_stock_level
** Description:
**                                                      Send a stock level request to the
server
** Parameters :
**                                                      void * - contex
**                                                      stockLevel_data_t * -
transaction data
** Returns                      :
**                                                      stockLevel_data_t * - return
buffer
** Comments                     :
**
*****************************************************************
*/

stockLevel_data_t * send_stock_level(void * contextP,stockLevel_data_t * dataP)
{
        trpc_status_t   trpcStatus;
        thread_info_t  *threadContex                    = (thread_info_t *) contextP;

        // common pre transaction call
        pre_rpc(threadContex,&dataP->header,STOCK_TRANS,0);

        // call stock level client interface
        dataP = tux_stock_level(threadContex,dataP,&trpcStatus);

        // common post transaction call
        post_rpc(threadContex,&dataP->header,STOCK_TRANS);

        return dataP;
}
```

tuxClientMonitor.hpp

```
////////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Project                      : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                                 : tuxClientMonitor.hpp
** Title                                : Tux Client Monitor
*****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                             :
**
*****************************************************************
*/

#ifndef __tuxClientMonitor_hpp__
#define __tuxClientMonitor_hpp__

#include <commonTuxDB2.hpp>
#include <tuxClientCommon.hpp>
#include <tuxClientInf.hpp>
#include <tuxClientInfUtilities.hpp>
#include <tuxClientThread.hpp>

////////////////////////////////////////////////////////////////////
// Definitions
////////////////////////////////////////////////////////////////////

#define MAX_CONSECUTIVE_ERRORS                  20

////////////////////////////////////////////////////////////////////
// Function Definitions
////////////////////////////////////////////////////////////////////

total_tran_count_t  *perfClntDataInit();

void start_bg_debug_thread(void);
void client_trace(char *comp,int value,int add);

void pre_rpc(thread_info_t *thread_infoP,data_header *headerP,int tran_type,int sub_tran_type);
void post_rpc(thread_info_t *thread_infoP,data_header *headerP,int tran_type);

void enroll_client(int user_id);
void init_tux_client(int user_id);
void *terminal_context_init(int fdIn);
void * clnt_thread_init();
void clnt_thread_done(void *contextP);
void exit_program(short err);

int too_many_errors(void * contextP);

newOrder_data_t * send_new_order(void * contextP,newOrder_data_t * dataP);
payment_data_t * send_payment(void * contextP,payment_data_t * dataP);
orderStatus_data_t * send_order_status(void * contextP,orderStatus_data_t * dataP);
delivery_data_t * send_delivery(void * contextP,delivery_data_t * dataP);
stockLevel_data_t * send_stock_level(void * contextP,stockLevel_data_t * dataP);

#endif
////////////////////////////////////////////////////////////////////
```

tuxClientNullDB.cpp

```
/*
*****************************************************************
** Project     : AIX DB/2 TPC-C
** Component   : TPC-C/Client
** Name        : tuxClientNullDB.cpp
** Title       : Abstraction layer for TUX client
*****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments     :
**
*****************************************************************
*/

#include <tuxClientNullDB.hpp>
```

```
////////////////////////////////////////////////////////////////
// Globals
////////////////////////////////////////////////////////////////

int             null_db                         = 0;

////////////////////////////////////////////////////////////////
// Null DB functions
////////////////////////////////////////////////////////////////

/*
****************************************************************
** Name       : tux_response
** Description :
**             Client interface to set transaction header
**             return codes
** Parameters  :
**             data_header - transaction data header
**             int - return code
**             int - sql_code
** Returns     :
**             None
** Comments    :
**
****************************************************************
*/

void tux_client_response(data_header *header,int rc,int sql_code)
{
        //set return codes
        header->dtype       = 0;
        header->returncode  = rc;
        header->sql_code    = sql_code;
        header->isam_code   = 0;
        header->num_rms     = 0;
        header->stats       = 0;

        return;
}

/*
****************************************************************
** Name       : tux_db_info_null_db
** Description :
**             Client interface to get Database Info
** Parameters  :
**             dbInfo_data_t * - transaction data
** Returns     :
**             None
** Comments    :
**
****************************************************************
*/

void tux_db_info_null_db(dbInfo_data_t * dataP)
{
        static INT16b dataSet;

        if (dataSet)
        {
                //set header return codes
                tux_client_response(&dataP->header,0,0);

                dataP->first_wh         =       10;
                dataP->last_wh                  =       5;
                dataP->server_id        =       2;

                dataSet=0;
        }
        else
        {
                //set header return codes
                tux_client_response(&dataP->header,0,0);

                dataP->first_wh         =       1000;
                dataP->last_wh                  =       678;
                dataP->server_id        =       10;

                dataSet=1;
        }

        return;
}

/*
****************************************************************
** Name       : tux_new_order_dump
** Description :
**             Client interface to dump transaction data
** Parameters  :
**                                                     FILE * - debug file ptr
**             newOrder_data_t * - transaction data
**                                                     int - flag to state which
transaction data to dump
** Returns     :
**             None
** Comments    :
**
****************************************************************
*/

void tux_new_order_dump(FILE *debugFile,newOrder_data_t * dataP,int flag)
{
        if(flag == 0)
        {
                fprintf(debugFile,"header->dtype:%d\n",dataP->header.dtype);
                fprintf(debugFile,"header->returncode:%d\n",dataP->header.returncode);
                fprintf(debugFile,"header->sql_code:%d\n",dataP->header.sql_code);
                fprintf(debugFile,"header->isam_code:%d\n",dataP->header.isam_code);
                fprintf(debugFile,"header->num_rms:%d\n",dataP->header.num_rms);
                fprintf(debugFile,"header->stats:%d\n",dataP->header.stats);
                fprintf(debugFile,"\n");
                fflush(debugFile);
        }
        else if(flag == 1)
        {
                fprintf(debugFile,"dataP->in.s_C_ID:%d\n",dataP->in.s_C_ID);
                fprintf(debugFile,"dataP->in.s_W_ID:%d\n",dataP->in.s_W_ID);
                fprintf(debugFile,"dataP->in.s_D_ID:%d\n",dataP->in.s_D_ID);
                fprintf(debugFile,"dataP->in.s_O_OL_CNT:%d\n",dataP->in.s_O_OL_CNT);
                fprintf(debugFile,"dataP->in.s_all_local:%d\n",dataP->in.s_all_local);

fprintf(debugFile,"dataP->in.duplicate_items:%d\n",dataP->in.duplicate_items);
                fprintf(debugFile,"\n");
                fflush(debugFile);
        }
        else if(flag == 2)
        {
                fprintf(debugFile,"dataP->out.s_C_LAST:%s\n",dataP->out.s_C_LAST);

fprintf(debugFile,"dataP->out.s_C_CREDIT:%s\n",dataP->out.s_C_CREDIT);
                fprintf(debugFile,"dataP->out.s_W_TAX:%d\n",dataP->out.s_W_TAX);

fprintf(debugFile,"dataP->out.s_C_DISCOUNT:%d\n",dataP->out.s_C_DISCOUNT);
                fprintf(debugFile,"dataP->out.s_O_ID:%d\n",dataP->out.s_O_ID);

fprintf(debugFile,"dataP->out.s_O_OL_CNT:%d\n",dataP->out.s_O_OL_CNT);

fprintf(debugFile,"dataP->out.s_total_amount:%d\n",dataP->out.s_total_amount);
                fprintf(debugFile,"dataP->out.s_transtatus:%d\n",dataP->out.s_transtatus);
                fprintf(debugFile,"\n");
                fflush(debugFile);
        }

        return;
}

/*
****************************************************************
** Name       : tux_new_order_null_db
** Description :
**             Client interface to do new order transaction
** Parameters  :
**             newOrder_data_t * - transaction data
** Returns     :
**             None
** Comments    :
**
****************************************************************
*/

void tux_new_order_null_db(newOrder_data_t * dataP)
{
        static INT16b dataSet;

        if (dataSet)
        {
                //set header return codes
                tux_client_response(&dataP->header,0,0);

                //data in
                dataP->in.s_C_ID                                        =
1150;
                dataP->in.s_W_ID                                        =
5;
                dataP->in.s_D_ID                                        =
8;
                dataP->in.s_O_OL_CNT                            =       4;
                dataP->in.s_all_local           =       1;
                dataP->in.duplicate_items       =       0;

                //data out
                strcpy(dataP->out.s_C_LAST,"QWERTQ");
                strcpy(dataP->out.s_C_CREDIT,"GC");
                dataP->out.s_W_TAX                              =
1694;
                dataP->out.s_D_TAX                              =
975;
                dataP->out.s_C_DISCOUNT         =       686;
                dataP->out.s_O_ID                                       =
3013;
                dataP->out.s_O_OL_CNT           =       4;
                dataP->out.s_total_amount       =       1135429;
                dataP->out.s_transtatus         =       0;


                //items
```

```
                strcpy(dataP->out.item[0].s_I_NAME,"QWERCBGQRBNMTAD");
                dataP->out.item[0].s_I_PRICE                    =
42000;
                dataP->out.item[0].s_OL_AMOUNT                  =
5540;
                dataP->out.item[0].s_S_QUANTITY                 =       31;
                dataP->out.item[0].s_brand_generic='G';

                strcpy(dataP->out.item[1].s_I_NAME,"WERRTYUIOP");
                dataP->out.item[1].s_I_PRICE                    =
45;
                dataP->out.item[1].s_OL_AMOUNT                  =
4381;
                dataP->out.item[1].s_S_QUANTITY                 =       57;
                dataP->out.item[1].s_brand_generic='G';

                strcpy(dataP->out.item[2].s_I_NAME,"ASDFGHJKLQ");
                dataP->out.item[2].s_I_PRICE                    =
673;
                dataP->out.item[2].s_OL_AMOUNT                  =
47173;
                dataP->out.item[2].s_S_QUANTITY                 =       42;

                dataP->out.item[2].s_brand_generic='G';

                strcpy(dataP->out.item[3].s_I_NAME,"QWERERTYUOOIM");
                dataP->out.item[3].s_I_PRICE                    =
350;
                dataP->out.item[3].s_OL_AMOUNT                  =
12096;
                dataP->out.item[3].s_S_QUANTITY                 =       84;

                dataP->out.item[3].s_brand_generic='G';

                dataSet=0;
        }
        else
        {
                //set header return codes
                tux_client_response(&dataP->header,0,0);

                //in
                dataP->in.s_C_ID                                =
599;
                dataP->in.s_W_ID                                =
10;
                dataP->in.s_D_ID                                =
4;
                dataP->in.s_O_OL_CNT                            =       9;
                dataP->in.s_all_local           =       1;
                dataP->in.duplicate_items       =       0;

                //out
                strcpy(dataP->out.s_C_LAST,"HJLKOPNB");
                strcpy(dataP->out.s_C_CREDIT,"GC");
                dataP->out.s_W_TAX                              =
913;
                dataP->out.s_D_TAX                              =
1519;
                dataP->out.s_C_DISCOUNT         =       958;
                dataP->out.s_O_ID                               =
1410;
                dataP->out.s_O_OL_CNT           =       9;
                dataP->out.s_total_amount       =       12345;
                dataP->out.s_transtatus         =       0;

                //items
                strcpy(dataP->out.item[0].s_I_NAME,"WERTYIUOPK");
                dataP->out.item[0].s_I_PRICE                    =
3000;
                dataP->out.item[0].s_OL_AMOUNT                  =
7696;
                dataP->out.item[0].s_S_QUANTITY                 =       97;
                dataP->out.item[0].s_brand_generic='G';

                strcpy(dataP->out.item[1].s_I_NAME,"WERTYOPKLD");
                dataP->out.item[1].s_I_PRICE                    =
100;
                dataP->out.item[1].s_OL_AMOUNT                  =
19299;
                dataP->out.item[1].s_S_QUANTITY                 =       51;
                dataP->out.item[1].s_brand_generic='G';

                strcpy(dataP->out.item[2].s_I_NAME,"WERTKJHGSX");
                dataP->out.item[2].s_I_PRICE                    =
400;
                dataP->out.item[2].s_OL_AMOUNT                  =
41670;
                dataP->out.item[2].s_S_QUANTITY                 =       14;
                dataP->out.item[2].s_brand_generic='G';

                strcpy(dataP->out.item[3].s_I_NAME,"DFGHJKLOPVB");
                dataP->out.item[3].s_I_PRICE                    =
14;
                dataP->out.item[3].s_OL_AMOUNT                  =
17213;
                dataP->out.item[3].s_S_QUANTITY                 =       90;
                dataP->out.item[3].s_brand_generic='G';

                strcpy(dataP->out.item[4].s_I_NAME,"DEFAULTTIEM");
                dataP->out.item[4].s_I_PRICE                    =
14;
                dataP->out.item[4].s_OL_AMOUNT                  =
27232;
                dataP->out.item[4].s_S_QUANTITY                 =       75;
                dataP->out.item[4].s_brand_generic='G';

                strcpy(dataP->out.item[5].s_I_NAME,"DFGULHIRNB");
                dataP->out.item[5].s_I_PRICE                    =
44;
                dataP->out.item[5].s_OL_AMOUNT                  =
35808;
                dataP->out.item[5].s_S_QUANTITY                 =       22;
                dataP->out.item[5].s_brand_generic='G';

                strcpy(dataP->out.item[6].s_I_NAME,"QWERTKJHLNV");
                dataP->out.item[6].s_I_PRICE                    =
55;
                dataP->out.item[6].s_OL_AMOUNT                  =
44392;
                dataP->out.item[6].s_S_QUANTITY                 =       21;
                dataP->out.item[6].s_brand_generic='G';

                strcpy(dataP->out.item[7].s_I_NAME,"DEFACULT");
                dataP->out.item[7].s_I_PRICE                    =
83;
                dataP->out.item[7].s_OL_AMOUNT                  =
8341;
                dataP->out.item[7].s_S_QUANTITY                 =       35;
                dataP->out.item[7].s_brand_generic='G';

                strcpy(dataP->out.item[8].s_I_NAME,"DEFAULT");
                dataP->out.item[8].s_I_PRICE                    =
100;
                dataP->out.item[8].s_OL_AMOUNT                  =
43160;
                dataP->out.item[8].s_S_QUANTITY                 =       73;
                dataP->out.item[8].s_brand_generic='G';

                dataSet=1;
        }
        return;
}

/*
****************************************************************
** Name       : tux_payment_dump
** Description :
**          Client interface to dump transaction data
** Parameters  :
**                                                              FILE * - debug file ptr
**          payment_data_t * - transaction data
**                                                              int - flag
** Returns     :
**          None
** Comments    :
**
****************************************************************
*/

void tux_payment_dump(FILE *debugFile,payment_data_t *dataP,int flag)
{
        if(flag==0)
        {
                fprintf(debugFile,"header->dtype:%d\n",dataP->header.dtype);
                fprintf(debugFile,"header->returncode:%d\n",dataP->header.returncode);
                fprintf(debugFile,"header->sql_code:%d\n",dataP->header.sql_code);
                fprintf(debugFile,"header->isam_code:%d\n",dataP->header.isam_code);
                fprintf(debugFile,"header->num_rms:%d\n",dataP->header.num_rms);
                fprintf(debugFile,"header->stats:%d\n",dataP->header.stats);
                fprintf(debugFile,"\n");
                fflush(debugFile);
        }
        else if(flag==1)
        {
fprintf(debugFile,"dataP->in.s_H_AMOUNT:%d\n",dataP->in.s_H_AMOUNT);
                fprintf(debugFile,"dataP->in.s_C_ID:%d\n",dataP->in.s_C_ID);
                fprintf(debugFile,"dataP->in.s_W_ID:%d\n",dataP->in.s_W_ID);
                fprintf(debugFile,"dataP->in.s_D_ID:%d\n",dataP->in.s_D_ID);
                fprintf(debugFile,"dataP->in.s_C_D_ID:%d\n",dataP->in.s_C_D_ID);
                fprintf(debugFile,"dataP->in.s_C_W_ID:%d\n",dataP->in.s_C_W_ID);

fprintf(debugFile,"dataP->in.s_H_DATE_time:%d\n",dataP->in.s_H_DATE_time);
                fprintf(debugFile,"dataP->in.s_C_LAST:%s\n",dataP->in.s_C_LAST);
                fprintf(debugFile,"\n");
                fflush(debugFile);
        }
        else if (flag==2)
        {
fprintf(debugFile,"dataP->out.s_C_CREDIT_LIM:%d\n",dataP->out.s_C_CREDIT_LIM);

fprintf(debugFile,"dataP->out.s_C_DISCOUNT:%d\n",dataP->out.s_C_DISCOUNT);

fprintf(debugFile,"dataP->out.s_C_BALANCE:%d\n",dataP->out.s_C_BALANCE);
                fprintf(debugFile,"dataP->out.s_C_ID:%d\n",dataP->out.s_C_ID);
```

```
fprintf(debugFile,"dataP->out.s_W_STREET_1:%s\n",dataP->out.s_W_STREET_1);

fprintf(debugFile,"dataP->out.s_W_STREET_2:%s\n",dataP->out.s_W_STREET_2);
                        fprintf(debugFile,"dataP->out.s_W_CITY:%s\n",dataP->out.s_W_CITY);
                        fprintf(debugFile,"dataP->out.s_W_STATE:%s\n",dataP->out.s_W_STATE);
                        fprintf(debugFile,"dataP->out.s_W_ZIP:%s\n",dataP->out.s_W_ZIP);
                        fprintf(debugFile,"dataP->out.s_C_FIRST:%s\n",dataP->out.s_C_FIRST);

fprintf(debugFile,"dataP->out.s_C_MIDDLE:%s\n",dataP->out.s_C_MIDDLE);
                        fprintf(debugFile,"dataP->out.s_C_LAST:%s\n",dataP->out.s_C_LAST);

fprintf(debugFile,"dataP->out.s_C_STREET_1:%s\n",dataP->out.s_C_STREET_1);

fprintf(debugFile,"dataP->out.s_C_STREET_2:%s\n",dataP->out.s_C_STREET_2);
                        fprintf(debugFile,"dataP->out.s_C_CITY:%s\n",dataP->out.s_C_CITY);
                        fprintf(debugFile,"dataP->out.s_C_STATE:%s\n",dataP->out.s_C_STATE);
                        fprintf(debugFile,"dataP->out.s_C_ZIP:%s\n",dataP->out.s_C_ZIP);
                        fprintf(debugFile,"dataP->out.s_C_PHONE:%s\n",dataP->out.s_C_PHONE);

fprintf(debugFile,"dataP->out.s_C_SINCE_time:%d\n",dataP->out.s_C_SINCE_time);

fprintf(debugFile,"dataP->out.s_C_CREDIT:%s\n",dataP->out.s_C_CREDIT);
                        fprintf(debugFile,"dataP->out.s_C_DATA:%s\n",dataP->out.s_C_DATA);
                        fprintf(debugFile,"\n");
                        fflush(debugFile);
        }

        return;
}


/*
****************************************************************
** Name       : tux_payment_null_db
** Description :
**        Client interface to do payment transaction
** Parameters  :
**        payment_data_t * - transaction data
** Returns    :
**        None
** Comments   :
**
****************************************************************
*/

void tux_payment_null_db(payment_data_t * dataP)
{
        static INT16b dataSet;

        if(dataSet)
        {
                //set header return codes
                tux_client_response(&dataP->header,0,0);

                dataP->header.dtype                             =       0;
                dataP->header.returncode        =       0;
                dataP->header.sql_code                  =       0;
                dataP->header.isam_code                 =       0;
                dataP->header.num_rms                   =       0;
                dataP->header.stats                             =       0;

                dataP->header.start_time.sec    =       0;
                dataP->header.start_time.usec           =       0;
                dataP->header.end_time.sec              =       0;
                dataP->header.end_time.usec     =       0;

                //in
                dataP->in.s_H_AMOUNT                            =
336397;
                dataP->in.s_C_ID
=       89;
                dataP->in.s_W_ID
=       7;
                dataP->in.s_D_ID
=       2;
                dataP->in.s_C_D_ID                                      =
7;
                dataP->in.s_C_W_ID                                      =
2;
                dataP->in.s_H_DATE_time                         =       0;
                strcpy(dataP->in.s_C_LAST,"Truck");

                //out

                dataP->out.s_C_CREDIT_LIM               =       5000000;
                dataP->out.s_C_DISCOUNT                 =       1024;
                dataP->out.s_C_BALANCE                  =       17815;
                dataP->out.s_C_ID                                       =
89;
                strcpy(dataP->out.s_W_STREET_1,"11501 Burnet Rd");
                strcpy(dataP->out.s_W_STREET_2,"BLD 905");
                strcpy(dataP->out.s_W_CITY,"Austin");
                strcpy(dataP->out.s_W_STATE,"TX");
                strcpy(dataP->out.s_W_ZIP,"78758");
                strcpy(dataP->out.s_D_STREET_1,"11900 Hobby Horse");
                strcpy(dataP->out.s_D_STREET_2,"Apt. 525");
                strcpy(dataP->out.s_D_CITY,"Valley");
                strcpy(dataP->out.s_D_STATE,"TX");
                strcpy(dataP->out.s_D_ZIP,"78559");
                strcpy(dataP->out.s_C_FIRST,"Jim");
                strcpy(dataP->out.s_C_MIDDLE,"F");
                strcpy(dataP->out.s_C_LAST,"Truck");
                strcpy(dataP->out.s_C_STREET_1,"100 N Solis");
                strcpy(dataP->out.s_C_STREET_2,"Flat 343");
                strcpy(dataP->out.s_C_CITY,"Cambridge");
                strcpy(dataP->out.s_C_STATE,"NY");
                strcpy(dataP->out.s_C_ZIP,"4562458976");
                strcpy(dataP->out.s_C_PHONE,"1234567890123456");
                dataP->out.s_C_SINCE_time=0;
                strcpy(dataP->out.s_C_CREDIT,"GC");
                strcpy(dataP->out.s_C_DATA,"Slow Payment-Bad Ebayer");
                dataSet=0;

        }
        else
        {
                //set header return codes
                tux_client_response(&dataP->header,0,0);

                //in
                dataP->in.s_H_AMOUNT            =       1725.02;
                dataP->in.s_C_ID                                        =
2986;
                dataP->in.s_W_ID                                        =
5;
                dataP->in.s_D_ID                                        =
2;
                dataP->in.s_C_D_ID                              =       2;
                dataP->in.s_C_W_ID                              =       10;
                dataP->in.s_H_DATE_time         =       0;
                strcpy(dataP->in.s_C_LAST,"Williams");

                //out
                dataP->out.s_C_CREDIT_LIM       =       40000;
                dataP->out.s_C_DISCOUNT         =       524;
                dataP->out.s_C_BALANCE          =       14080;
                dataP->out.s_C_ID                                       =
2986;
                strcpy(dataP->out.s_W_STREET_1,"1201 Park Ave");
                strcpy(dataP->out.s_W_STREET_2,"Suite 432");
                strcpy(dataP->out.s_W_CITY,"Denver");
                strcpy(dataP->out.s_W_STATE,"CO");
                strcpy(dataP->out.s_W_ZIP,"787562356");
                strcpy(dataP->out.s_D_STREET_1,"3404 Garth Rd");
                strcpy(dataP->out.s_D_STREET_2,"Suite 320");
                strcpy(dataP->out.s_D_CITY,"Austin");
                strcpy(dataP->out.s_D_STATE,"TX");
                strcpy(dataP->out.s_D_ZIP,"785599876");
                strcpy(dataP->out.s_C_FIRST,"John");
                strcpy(dataP->out.s_C_MIDDLE,"P");
                strcpy(dataP->out.s_C_LAST,"Williams");
                strcpy(dataP->out.s_C_STREET_1,"North Rab Road");
                strcpy(dataP->out.s_C_STREET_2,"Apt 343");
                strcpy(dataP->out.s_C_CITY,"La Grange");
                strcpy(dataP->out.s_C_STATE,"TX");
                strcpy(dataP->out.s_C_ZIP,"785585432");
                strcpy(dataP->out.s_C_PHONE,"9876543210987654");
                dataP->out.s_C_SINCE_time=0;
                strcpy(dataP->out.s_C_CREDIT,"GC");
                strcpy(dataP->out.s_C_DATA,"HJLGOPUSN");

                dataSet=1;
        }

        return;
}

/*
****************************************************************
** Name       : tux_order_status_dump
** Description :
**        Client interface to dump transaction data
** Parameters  :
**                                                      FILE * - debug file ptr
**        newOrder_data_t * - transaction data
**                                                      int - flag to state which
transaction data to dump
** Returns    :
**        None
** Comments   :
**
****************************************************************
*/

void tux_order_status_dump(FILE *debugFile,orderStatus_data_t * dataP,int flag)
{
        if(flag == 0)
        {
                fprintf(debugFile,"header->dtype:%d\n",dataP->header.dtype);
                fprintf(debugFile,"header->returncode:%d\n",dataP->header.returncode);
                fprintf(debugFile,"header->sql_code:%d\n",dataP->header.sql_code);
                fprintf(debugFile,"header->isam_code:%d\n",dataP->header.isam_code);
                fprintf(debugFile,"header->num_rms:%d\n",dataP->header.num_rms);
                fprintf(debugFile,"header->stats:%d\n",dataP->header.stats);
                fprintf(debugFile,"\n");
                fflush(debugFile);

        }
```

```
		else if (flag == 1)
		{
			fprintf(debugFile,"dataP->in.s_C_ID:%d\n",dataP->in.s_C_ID);
			fprintf(debugFile,"dataP->in.s_W_ID:%d\n",dataP->in.s_W_ID);
			fprintf(debugFile,"dataP->in.s_D_ID:%d\n",dataP->in.s_D_ID);
			fprintf(debugFile,"dataP->in.s_C_LAST:%s\n",dataP->in.s_C_LAST);
			fprintf(debugFile,"\n");
			fflush(debugFile);
		}
		else if (flag == 2)
		{
fprintf(debugFile,"dataP->out.s_C_BALANCE:%d\n",dataP->out.s_C_BALANCE);
			fprintf(debugFile,"dataP->out.s_C_ID:%d\n",dataP->out.s_C_ID);
			fprintf(debugFile,"dataP->out.s_O_ID:%d\n",dataP->out.s_O_ID);
fprintf(debugFile,"dataP->out.s_O_CARRIER_ID:%d\n",dataP->out.s_O_CARRIER_ID);
			fprintf(debugFile,"dataP->out.s_ol_cnt:%d\n",dataP->out.s_ol_cnt);
			fprintf(debugFile,"dataP->out.s_C_FIRST:%s\n",dataP->out.s_C_FIRST);
fprintf(debugFile,"dataP->out.s_C_MIDDLE:%s\n",dataP->out.s_C_MIDDLE);
			fprintf(debugFile,"dataP->out.s_C_LAST:%s\n",dataP->out.s_C_LAST);
			fprintf(debugFile,"dataP->out.s_transtatus:%d\n",dataP->out.s_transtatus);
			fprintf(debugFile,"dataP->out.deadlocks:%d\n",dataP->out.deadlocks);
			fprintf(debugFile,"\n");
			fflush(debugFile);
		}
		return;
}

/*
****************************************************************
** Name       : tux_order_status_null_db
** Description :
**		Client interface to do order status transaction
** Parameters  :
**		orderStatus_data_t * - transaction data
** Returns     :
**		None
** Comments    :
**
****************************************************************
*/

void tux_order_status_null_db(orderStatus_data_t * dataP)
{
		static INT16b dataSet;

		if (dataSet)
		{
			//set header return codes
			tux_client_response(&dataP->header,0,0);

			//in
			dataP->in.s_C_ID                        =
335;
			dataP->in.s_W_ID                        =
9;
			dataP->in.s_D_ID                        =
3;
			strcpy(dataP->in.s_C_LAST,"Simpson");


			//out
			dataP->out.s_C_BALANCE          =       1000;
			dataP->out.s_C_ID                       =
3;
			dataP->out.s_O_ID                       =
1696;
			dataP->out.s_O_CARRIER_ID       =       9;
			dataP->out.s_ol_cnt                     =       6;
			strcpy(dataP->out.s_C_FIRST,"Homer");
			strcpy(dataP->out.s_C_MIDDLE,"J");
			strcpy(dataP->out.s_C_LAST,"Simpson");
			dataP->out.s_transtatus         =       0;
			dataP->out.deadlocks                    =       0;

			//items
			dataP->out.item[0].s_OL_AMOUNT          =
300;
			dataP->out.item[0].s_OL_I_ID                    =
23492;
			dataP->out.item[0].s_OL_SUPPLY_W_ID     =       9;
			dataP->out.item[0].s_OL_QUANTITY        =       5;
			dataP->out.item[0].s_OL_DELIVERY_D_time=1234567890;

			dataP->out.item[1].s_OL_AMOUNT          =
123;
			dataP->out.item[1].s_OL_I_ID                    =
18860;
			dataP->out.item[1].s_OL_SUPPLY_W_ID     =       9;
			dataP->out.item[1].s_OL_QUANTITY        =       5;
			dataP->out.item[1].s_OL_DELIVERY_D_time=1234567890;

			dataP->out.item[2].s_OL_AMOUNT          =
150;
			dataP->out.item[2].s_OL_I_ID                    =
90488;
			dataP->out.item[2].s_OL_SUPPLY_W_ID     =       9;
			dataP->out.item[2].s_OL_QUANTITY        =       5;
			dataP->out.item[2].s_OL_DELIVERY_D_time=1234567890;

			dataP->out.item[3].s_OL_AMOUNT          =
250;
			dataP->out.item[3].s_OL_I_ID                    =
22741;
			dataP->out.item[3].s_OL_SUPPLY_W_ID     =       9;
			dataP->out.item[3].s_OL_QUANTITY        =       5;
			dataP->out.item[3].s_OL_DELIVERY_D_time=1234567890;

			dataP->out.item[4].s_OL_AMOUNT          =
200;
			dataP->out.item[4].s_OL_I_ID                    =
92952;
			dataP->out.item[4].s_OL_SUPPLY_W_ID     =       9;
			dataP->out.item[4].s_OL_QUANTITY        =       5;
			dataP->out.item[4].s_OL_DELIVERY_D_time=1234567890;

			dataP->out.item[5].s_OL_AMOUNT          =
2345;
			dataP->out.item[5].s_OL_I_ID                    =
29956;
			dataP->out.item[5].s_OL_SUPPLY_W_ID     =       9;
			dataP->out.item[5].s_OL_QUANTITY        =       5;
			dataP->out.item[5].s_OL_DELIVERY_D_time=1234567890;

			dataSet=0;
		}
		else
		{
			//set header return codes
			tux_client_response(&dataP->header,0,0);

			//in
			dataP->in.s_C_ID                        =
856;
			dataP->in.s_W_ID                        =
10;
			dataP->in.s_D_ID                        =
2;
			strcpy(dataP->in.s_C_LAST,"Forman");

			//out
			dataP->out.s_C_BALANCE          =       1000;
			dataP->out.s_C_ID                       =
856;
			dataP->out.s_O_ID                       =
418;
			dataP->out.s_O_CARRIER_ID       =       10;
			dataP->out.s_ol_cnt                     =       5;
			strcpy(dataP->out.s_C_FIRST,"Erick");
			strcpy(dataP->out.s_C_MIDDLE,"J");
			strcpy(dataP->out.s_C_LAST,"Forman");
			dataP->out.s_transtatus         =       0;

			//items
			dataP->out.item[0].s_OL_AMOUNT          =
120;
			dataP->out.item[0].s_OL_I_ID                    =
54602;
			dataP->out.item[0].s_OL_SUPPLY_W_ID     =       10;
			dataP->out.item[0].s_OL_QUANTITY        =       5;
			dataP->out.item[0].s_OL_DELIVERY_D_time=1234567890;

			dataP->out.item[1].s_OL_AMOUNT          =       23;
			dataP->out.item[1].s_OL_I_ID                    =
18860;
			dataP->out.item[1].s_OL_SUPPLY_W_ID     =       10;
			dataP->out.item[1].s_OL_QUANTITY        =       5;
			dataP->out.item[1].s_OL_DELIVERY_D_time=1234567890;

			dataP->out.item[2].s_OL_AMOUNT          =
560;
			dataP->out.item[2].s_OL_I_ID                    =
90488;
			dataP->out.item[2].s_OL_SUPPLY_W_ID     =       10;
			dataP->out.item[2].s_OL_QUANTITY        =       5;
			dataP->out.item[2].s_OL_DELIVERY_D_time=1234567890;

			dataP->out.item[3].s_OL_AMOUNT          =
980;
			dataP->out.item[3].s_OL_I_ID                    =
22741;
			dataP->out.item[3].s_OL_SUPPLY_W_ID     =       10;
			dataP->out.item[3].s_OL_QUANTITY        =       5;
			dataP->out.item[3].s_OL_DELIVERY_D_time=1234567890;

			dataP->out.item[4].s_OL_AMOUNT          =
250;
			dataP->out.item[4].s_OL_I_ID                    =
92952;
			dataP->out.item[4].s_OL_SUPPLY_W_ID     =       10;
			dataP->out.item[4].s_OL_QUANTITY        =       5;
			dataP->out.item[4].s_OL_DELIVERY_D_time=1234567890;

			dataSet=1;
		}
```

```
        return;
}
/*
****************************************************************
** Name       : tux_delievery_dump
** Description :
**         Client interface to dump transaction data
** Parameters  :
**                                                         FILE * - debug file ptr
**         delivery_data_t * - transaction data
**                                                         int - flag to state which
transaction data to dump
** Returns     :
**         None
** Comments    :
**
****************************************************************
*/

void tux_delievery_dump(FILE *debugFile,delivery_data_t * dataP,int flag)
{
        if(flag==0)
        {
                fprintf(debugFile,"header->dtype:%d\n",dataP->header.dtype);
                fprintf(debugFile,"header->returncode:%d\n",dataP->header.returncode);
                fprintf(debugFile,"header->sql_code:%d\n",dataP->header.sql_code);
                fprintf(debugFile,"header->isam_code:%d\n",dataP->header.isam_code);
                fprintf(debugFile,"header->num_rms:%d\n",dataP->header.num_rms);
                fprintf(debugFile,"header->stats:%d\n",dataP->header.stats);
                fprintf(debugFile,"\n");
                fflush(debugFile);
        }
        else if(flag==1)
        {
                fprintf(debugFile,"dataP->in.s_W_ID:%d\n",dataP->in.s_W_ID);

fprintf(debugFile,"dataP->in.s_O_CARRIER_ID:%d\n",dataP->in.s_O_CARRIER_ID );

fprintf(debugFile,"dataP->in.s_O_DELIVERY_D_time:%d\n",dataP->in.s_O_DELIVERY_D_time);
                fprintf(debugFile,"\n");
                fflush(debugFile);
        }
        else if (flag==2)
        {
                fprintf(debugFile,"dataP->out.s_transtatus:%d\n",dataP->out.s_transtatus);
                fprintf(debugFile,"dataP->out.deadlocks:%d\n",dataP->out.deadlocks);
                fprintf(debugFile,"dataP->out.s_O_ID[0]:%d\n",dataP->out.s_O_ID[0]);
                fprintf(debugFile,"\n");
                fflush(debugFile);
        }

        return;
}
/*
****************************************************************
** Name       : tux_delivery_null_db
** Description :
**         Client interface to do delivery transaction
** Parameters  :
**         delivery_data_t * - transaction data
** Returns     :
**         None
** Comments    :
**
****************************************************************
*/

void tux_delivery_null_db(delivery_data_t * dataP)
{
        static INT16b dataSet;

        if(dataSet)
        {
                //set header return codes
                tux_client_response(&dataP->header,0,0);

                //in
                dataP->in.s_W_ID                = 8;
                dataP->in.s_O_CARRIER_ID                = 5;
                dataP->in.s_O_DELIVERY_D_time   = 0;

                //out
                dataP->out.s_transtatus         =       0;
                dataP->out.deadlocks            =       0;
                dataP->out.s_O_ID[0]            =       0;

                dataSet=0;
        }
        else
        {
                //set header return codes
                tux_client_response(&dataP->header,0,0);


                //in
                dataP->in.s_W_ID                = 7;
                dataP->in.s_O_CARRIER_ID                = 2;
                dataP->in.s_O_DELIVERY_D_time   = 0;

                //out
                dataP->out.s_transtatus                 =
0;
                dataP->out.deadlocks
=               0;
                dataP->out.s_O_ID[0]
=               0;

                dataSet=1;
        }

        return;
}

/*
****************************************************************
** Name       : tux_stock_level_dump
** Description :
**         Client interface to dump transaction data
** Parameters  :
**                                                         FILE * - debug file ptr
**         stockLevel_data_t * - transaction data
**                                                         int - flag to state which
transaction data to dump
** Returns     :
**         None
** Comments    :
**
****************************************************************
*/

void tux_stock_level_dump(FILE * debugFile,stockLevel_data_t * dataP,int flag)
{
        if(flag==0)
        {
                fprintf(debugFile,"header->dtype:%d\n",dataP->header.dtype);
                fprintf(debugFile,"header->returncode:%d\n",dataP->header.returncode);
                fprintf(debugFile,"header->sql_code:%d\n",dataP->header.sql_code);
                fprintf(debugFile,"header->isam_code:%d\n",dataP->header.isam_code);
                fprintf(debugFile,"header->num_rms:%d\n",dataP->header.num_rms);
                fprintf(debugFile,"header->stats:%d\n",dataP->header.stats);
                fprintf(debugFile,"\n");
                fflush(debugFile);
        }
        else if(flag==1)
        {
                fprintf(debugFile,"dataP->in.s_W_ID:%d\n",dataP->in.s_W_ID);
                fprintf(debugFile,"dataP->in.s_D_ID:%d\n",dataP->in.s_D_ID);
                fprintf(debugFile,"dataP->in.s_threshold:%d\n");
                fprintf(debugFile,"\n");
                fflush(debugFile);
        }
        else if(flag==2)
        {
                fprintf(debugFile,"dataP->out.s_transtatus:%d\n",dataP->out.s_transtatus);
                fprintf(debugFile,"dataP->out.deadlocks:%d\n",dataP->out.deadlocks);
                fprintf(debugFile,"dataP->out.s_low_stock:%d\n",dataP->out.s_low_stock);
                fprintf(debugFile,"\n");
                fflush(debugFile);
        }

        return;
}

/*
****************************************************************
** Name       : tux_stock_level_null_db
** Description :
**         Client interface to do stock level transaction
** Parameters  :
**         stockLevel_data_t * - transaction data
** Returns     :
**         None
** Comments    :
**
****************************************************************
*/

void tux_stock_level_null_db(stockLevel_data_t * dataP)
{
        static INT16b dataSet;

        if (dataSet)
        {
                //set header return codes
                tux_client_response(&dataP->header,0,0);

                //in
                dataP->in.s_W_ID
=               9;
                dataP->in.s_D_ID        = 5;
                dataP->in.s_threshold   = 10;

                //out
                dataP->out.s_transtatus                 =               0;
                dataP->out.deadlocks                    =
0;
```

```
                        dataP->out.s_low_stock                          =
100;

                        dataSet=0;
                }
                else
                {
                        //set header return codes
                        tux_client_response(&dataP->header,0,0);

                        //in
                        dataP->in.s_W_ID
=               6;
                        dataP->in.s_D_ID        = 7;
                        dataP->in.s_threshold   = 20;

                        //out
                        dataP->out.s_transtatus                         =               0;
                        dataP->out.deadlocks                            =
0;
                        dataP->out.s_low_stock                          =               50;

                        dataSet=1;
                }

                return;

}
```

tuxClientNullDB.hpp

```
/*
************************************************************************
** Project     : AIX DB/2 TPC-C
** Component   : TPC-C/Client
** Name        : tuxClientNullDB.hpp
** Title       : Terminal control
************************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History       :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments      :
**
************************************************************************
*/

#ifndef __tuxClientNullDB_hpp__
#define __tuxClientNullDB_hpp__

#include <commonTuxDB2.hpp>

//////////////////////////////////////////////////////////////////
// Externals
//////////////////////////////////////////////////////////////////

extern int      null_db;

//////////////////////////////////////////////////////////////////
// Function Definitions
//////////////////////////////////////////////////////////////////

void tux_db_info_null_db(dbInfo_data_t * dataP);
void tux_new_order_dump(FILE *debugFile,newOrder_data_t * dataP,int flag);
void tux_payment_dump(FILE *debugFile,payment_data_t * dataP,int flag);
void tux_order_status_dump(FILE *debugFile,orderStatus_data_t * dataP,int flag);
void tux_delivery_dump(FILE *debugFile,delivery_data_t * dataP,int flag);
void tux_stock_level_dump(FILE *debugFile,stockLevel_data_t * dataP,int flag);

// TPCC transactions
void tux_new_order_null_db(newOrder_data_t * dataP);
void tux_payment_null_db(payment_data_t * dataP);
void tux_order_status_null_db(orderStatus_data_t * dataP);
void tux_delivery_null_db(delivery_data_t * dataP);
void tux_stock_level_null_db(stockLevel_data_t * dataP);

#endif
//////////////////////////////////////////////////////////////////
```

tuxClientScreen.cpp

```
/*
************************************************************************
** Project                      : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                         : tuxClientScreen.cpp
** Title                        : Tux client terminal
************************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History       :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                     :
**
************************************************************************
*/

#include <tuxClientScreen.hpp>

//////////////////////////////////////////////////////////////////
// Statics
//////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////
// Screen class
//////////////////////////////////////////////////////////////////

char            Screen::end_str[]                               = "\033[H\n";
int             Screen::end_str_len             = sizeof(Screen::end_str) - 1;

//////////////////////////////////////////////////////////////////
// NewOrder class
//////////////////////////////////////////////////////////////////

char 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 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 class
////////////////////////////////////////////////////////////////

char 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 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;

////////////////////////////////////////////////////////////////
// OrderStatus class
////////////////////////////////////////////////////////////////

char 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 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 class
////////////////////////////////////////////////////////////////

char Delivery::static_screen[] =
                POS( 1, 3) CLEAR_EOS
                POS(38, 3) "Delivery"
                POS( 1, 4) "Warehouse:"
                POS( 1, 6) "Carrier Number:"
;

char 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;

////////////////////////////////////////////////////////////////
// StockLevel class
////////////////////////////////////////////////////////////////

char 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 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 class
////////////////////////////////////////////////////////////////

char Login::static_screen[] =
                POS( 1, 1) CLEAR_EOS
                POS(30, 3) "Please login."
                POS( 5, 5) "Warehouse:"
                POS(24, 5) "District:"
;

char 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 class
////////////////////////////////////////////////////////////////

char 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;

////////////////////////////////////////////////////////////////
// Class functions
////////////////////////////////////////////////////////////////

////////////////////////////////////////////////////////////////
// Screen class
////////////////////////////////////////////////////////////////
```

```
/*
*********************************************************************
** Name                                 : Screen::reset
** Description:
**                                                                    Reset screen
** Parameters :
**                                                                    None
** Returns                      :
**                                                                    int - return code
** Comments                     :
**
*********************************************************************
*/

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;
}

/*
*********************************************************************
** Name                                 : Screen::present_empty_fields
** Description:
**                                                                    Clear fields?
** Parameters :
**                                                                    None
** Returns                      :
**                                                                    int - return code
** Comments                     :
**
*********************************************************************
*/

int Screen::present_empty_fields()
{
        if (empty_fields)
                threadP->write(empty_fields,empty_fields_len);

        return 0;
}

/*
*********************************************************************
** Name                                 : Screen::present
** Description:
**                                                                    Show field
** Parameters :
**                                                                    None
** Returns                      :
**                                                                    int - return code
** Comments                     :
**
*********************************************************************
*/

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);
                }
        }
        else
        {
                present_empty_fields();
        }

        return 0;
}

/*
*********************************************************************
** Name                                 : Screen::user_input
** Description:
**                                                                    Enter field values
** Parameters :
**                                                                    None
** Returns                      :
**                                                                    int - return code
** Comments                     :
**
*********************************************************************
*/

int Screen::user_input()
{
        int             key;

        has_data = 1;

        fields[pos]->start_position();
        threadP->flush();

        // get field values
        key = fields[pos]->get_field(0);
        do
        {
                switch(key)
                {
                case EOF:
                        return 0;
                case Field::NEXT_FIELD:
                        if(fields[++pos] == NULL)
                                pos = 0;

                        // some fields only take an entry if the previous field contains
no entry
                        if(fields[pos]->blank_with_prev())
                        {
                                if(fields[pos - 1]->pos)
                                {
                                        if(fields[++pos] == NULL)
                                                pos = 0;
                                }
                        }
                        break;
                case Field::PREV_FIELD:
                        if(--pos < 0)
                        {
                                while(fields[++pos] != NULL);
                                pos--;
                        }

                        // some fields only take an entry if the previous field contains
no entry
                        if(fields[pos]->blank_with_prev())
                        {
                                if(fields[pos - 1]->pos)
                                        pos--;
                        }
                        break;
                case Field::REDISPLAY:
                        present();
                        break;
                case Field::ABORT:
                        threadP->write(end_str,end_str_len);
                        return 0;
                case Field::ENTER:
                        if (validate())
                                return 1;
                        break;
                }

                key = fields[pos]->get_field();
        } while (TRUE);

        // we don't get here
        return 0;
}

/*
*********************************************************************
** Name                                 : Screen::display_status
** Description:
**                                                                    Present Status
** Parameters :
**                                                                    int - status code
** Returns                      :
**                                                                    int - return code
** Comments                     :
**
*********************************************************************
*/

int Screen::display_status(int status)
{
        position(threadP,status_x,status_y);
        threadP->write("Execution Status: ");

        // check status code
        switch(status)
        {
        case OK:
                threadP->write("Transaction Committed");
                break;
        case INVALID_ITEM:
                threadP->write("Item number is not valid");
                break;
        case RPC_ERROR:
                threadP->write("ERROR: Rollback -- BAD RPC");
                break;
        default:
                {
                        threadP->write("ERROR: Rollback -- ");
```

```
                                        char buf[6];
                                        format_int(buf,6,status);
                                        threadP->write(buf,5);
                                }
                        }
                        return 0;
}

/*
****************************************************************
** Name                                 : Screen::handle
** Description:
**                                                              Handle screen event?
** Parameters :
**                                                              None
** Returns              :
**                                                              int - return code
** Comments             :
**
****************************************************************
*/

int Screen::handle()
{
                reset();

                threadP->hold(); present();

                threadP->write(TRIGGER,1);

                if(!user_input())
                {
                        threadP->write(end_str,end_str_len);
                        threadP->write(TRIGGER,1);
                        return ERR;
                }

                threadP->flush(); threadP->hold();

                if (process())
                {
                        threadP->write(end_str,end_str_len);
                        threadP->write(TRIGGER,1);
                        return ERR;
                }

                respond();

                threadP->write(end_str,end_str_len);
                threadP->write(TRIGGER,1);
                threadP->flush();

                return OK;
}
/*
****************************************************************
** Name                                 : Screen::process
** Description:
**                                                              Run transaction place holder
** Parameters :
**                                                              None
** Returns              :
**                                                              int - return code
** Comments             :
**
****************************************************************
*/

int Screen::process()
{
                return 0;
}
/*
****************************************************************
** Name                                 : Screen::update_address
** Description:
**                                                              Handle a change in the buffer
address change
** Parameters :
**                                                              void * - new buffer address
** Returns              :
**                                                              None
** Comments             :
**                                                              We rplace the old address with
the new address,
**                                                              cleanup of the allocated
memory has already been
**                                                              done by the time we find out
about the change!
**
****************************************************************
*/

void Screen::update_address(void *bufferAddress)
{
                void * oldPtr = dataptr;
                int adjustment = (char *) bufferAddress - (char *) dataptr;
                if(!adjustment) return;

                if(fields != NULL)
                {
                        for(int lpos = 0;fields[lpos] != NULL;lpos++)
                                fields[lpos]->update_field_address(adjustment);
                }

                return;
}

/*
****************************************************************
** Name                                 : Screen::~Screen
** Description:
**                                                              Destructor
** Parameters :
**                                                              None
** Returns              :
**                                                              None
** Comments             :
**
****************************************************************
*/

Screen::~Screen()
{
                if(fields != NULL)
                {
                        for(int lpos = 0;fields[lpos] != NULL;lpos++)
                                delete fields[lpos];

                        delete [] fields;
                }

                fields = NULL;

                if(dataptr)
                        tuxBufferDeallocate(dataptr);

                return;
}

//////////////////////////////////////////////////////////////////
// NewOrder class
//////////////////////////////////////////////////////////////////

/*
****************************************************************
** Name                                 : NewOrder::reset
** Description:
**                                                              Reset fields
** Parameters :
**                                                              None
** Returns              :
**                                                              int - return code
** Comments             :
**
****************************************************************
*/

int NewOrder::reset()
{
  Screen::reset();

  pos = start_field;
  memset(dataptr,0,sizeof(*data));

  return 0;
}


/*
****************************************************************
** Name                                 : NewOrder::NewOrder
** Description:
**                                                              Constructor
** Parameters :
**                                                              User_data * - User
**                                                              Thread_data * - Thread
** Returns              :
**                                                              None
** Comments             :
**
****************************************************************
*/

NewOrder::NewOrder(User_data *udP,Thread_data *threadP) : Screen(udP,threadP)
{
                data                                    = (newOrder_data_t *)
tuxBufferAllocate(sizeof(newOrder_data_t));

                tran_type               = NEW_ORDER_TRANS;
                data_len                        = sizeof(newOrder_data_t);
                dataptr                         = 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->in.in_item[i].s_OL_SUPPLY_W_ID);
                fields[lpos++] = genfield(threadP,10,9+i,6,&data->in.in_item[i].s_OL_I_ID);
                fields[lpos++] =
genfield(threadP,45,9+i,2,&data->in.in_item[i].s_OL_QUANTITY);
        }

        start_field             = lpos;
        fields[lpos++] = genfield(threadP,29,4,4,&data->in.s_D_ID);      // District
        fields[lpos++] = genfield(threadP,12,5,4,&data->in.s_C_ID);      // Customer
        fields[lpos++] = NULL;

        reset(); return;
}
/*
*****************************************************************
** Name                         : NewOrder::validate
** Description:
**                                                                 Validate entries
** Parameters :
**                                                                 None
** Returns              :
**                                                                 int - Return code
** Comments             :
**
*****************************************************************
*/

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

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

        data->in.s_O_OL_CNT             = 0;
        data->in.s_all_local            = 1;
        data->in.s_W_ID                         = user_dataP->warehouse;

        // check each field
        int last = -1;    int i = 0;
        for(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;
                                message(threadP,"Warehouse ID is a required
field");
                                return 0;
                        }

                        if(!fields[i+1]->pos)
                        {
                                pos = i + 1;
                                message(threadP,"Item ID is a required field");
                                return 0;
                        }

                        if(data->in.in_item[i/3].s_OL_QUANTITY <= 0)
                        {
                                pos = i + 2;
                                message(threadP,"Please enter a quantity greater
than 0");
                                return 0;
                        }
                        if(data->in.in_item[i/3].s_OL_SUPPLY_W_ID !=
data->in.s_W_ID)
                                data->in.s_all_local = 0;

                        data->in.s_O_OL_CNT++;
                }
                else
                {
                        if (last < 0)
                                last = i;
                }
        }

        // validate item
        if (data->in.s_O_OL_CNT <= 0)
        {
                pos = 0;
                message(threadP,"Please enter an item to order");
                return 0;

        }

        //  Compress the order lines:  some of them may be empty

        int ind;
        for(i = 0,ind = 0;ind < data->in.s_O_OL_CNT;i++)
        {
                if(fields[i*3]->pos)
                {
                        if(i > ind)
                                data->in.in_item[ind] = data->in.in_item[i];

                        ind ++;
                }
        }

        // skipped order lines
        if(i > ind)
        {
                int j;
                for(j = ind;j < i;j++)
                {
                        //  At least one empty line was skipped
                        data->in.in_item[j].s_OL_SUPPLY_W_ID  = 0;
                        data->in.in_item[j].s_OL_I_ID
        = 0;
                        data->in.in_item[j].s_OL_QUANTITY                       =
0;
                }
        }

        return 1;
}

/*
*****************************************************************
** Name                         : NewOrder::respond
** Description:
**                                                                 Display values
** Parameters :
**                                                                 None
** Returns              :
**                                                                 int - Return code
** Comments             :
**
*****************************************************************
*/

int NewOrder::respond()
{
        int                     i;

        double          cost;
        double          amount;
        double          total_amount;

        char                    buf[32];

        position(threadP, 1,9);         clear_eos(threadP);
        position(threadP,25,5);         threadP->write(data->out.s_C_LAST);
        position(threadP,52,5);         threadP->write(data->out.s_C_CREDIT);
        position(threadP,15,6);         format_int(buf,10,data->out.s_O_ID);

threadP->write(buf,9);

  // check transaction return code
        display_status(data->header.sql_code);
        // check the return code from tuxedo tpcall
        if (data->header.sql_code != OK)
        {
                #if CHECK_RC
                if (data->header.sql_code < OK)
                {
                        fprintf(stderr,"NEW ORDER FAILURE, s_transtatus:%d
s_O_ID:%d\n",data->header.sql_code,data->out.s_O_ID);
                }
                #endif
                return ERR;
        }
```

```
		position(threadP,25,5);		threadP->write(data->out.s_C_LAST);
		position(threadP,52,5);		threadP->write(data->out.s_C_CREDIT);
		position(threadP,48,6);		format_int(buf,3,data->out.s_O_OL_CNT);
									threadP->write(buf,2);
		position(threadP,61,4);		format_datetime(buf,20,data->out.s_O_ENTRY_D_time);
				threadP->write(buf,19);
		position(threadP,64,5);
format_float(buf,6,2,(double)(data->out.s_C_DISCOUNT/100.0));
		threadP->write(buf,5);
		position(threadP,59,6);		format_float(buf,6,2,(double)(data->out.s_W_TAX/100.0));

threadP->write(buf,5);
		position(threadP,74,6);		format_float(buf,6,2,(double)(data->out.s_D_TAX/100.0));
									threadP->write(buf,5);

		total_amount = 0;
		for (i=0;i < data->out.s_O_OL_CNT;i++)
		{
				position(threadP, 3,9 + i);
format_int(buf,6,data->in.in_item[i].s_OL_SUPPLY_W_ID);		threadP->write(buf,5);
				position(threadP,10,9 + i);
format_int(buf,7,data->in.in_item[i].s_OL_I_ID);
threadP->write(buf,6);
				position(threadP,19,9 + i);	threadP->write(data->out.item[i].s_I_NAME);
				position(threadP,45,9 + i);
format_int(buf,3,data->in.in_item[i].s_OL_QUANTITY);		threadP->write(buf,2);
				position(threadP,51,9 + i);
format_int(buf,4,data->out.item[i].s_S_QUANTITY);
threadP->write(buf,3);
				position(threadP,58,9 + i);
threadP->write(&data->out.item[i].s_brand_generic,1);
				position(threadP,62,9 + i);
format_money(buf,8,(double)(data->out.item[i].s_I_PRICE/100.0));
threadP->write(buf,7);
				position(threadP,71,9 + i);
format_money(buf,10,(double)(data->out.item[i].s_OL_AMOUNT/100.0));
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,(double)(data->out.s_total_amount/100.0));
		threadP->write(buf,9);

		return OK;
}

/*
*****************************************************************
** Name				: NewOrder::process
** Description:
**							Run transaction
** Parameters :
**							None
** Returns		:
**							int - return code
** Comments		:
**
*****************************************************************
*/

int NewOrder::process()
{
		if(tran_type != NEW_ORDER_TRANS)
				return 0;

		newOrder_data_t *oldPtr = data;

		data = send_new_order(threadP->contextP,data);

		newOrder_data_t *newPtr = data;

		update_address((void *) data);

		return 0;
}

////////////////////////////////////////////////////////////////////////
// Payment class
////////////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name				: Payment::Payment
** Description:
**							Constructor
** Parameters :
**							User_data * - User
**							Thread_data * - Thread
** Returns		:
**							None
** Comments		:
**
*****************************************************************
*/

Payment::Payment(User_data *udP,Thread_data *threadP) : Screen(udP,threadP)
{
		data                                    = (payment_data_t *)
tuxBufferAllocate(sizeof(payment_data_t));

		tran_type               = PAYMENT_TRANS;
		data_len                                = sizeof(payment_data_t);
		dataptr                                 = data;

		status_x      = 1;
  status_y      = 19;

		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;

		int lpos = 0; fields = new Field *[7];
		fields[lpos++] = genfield(threadP,52, 6, 2,&data->in.s_D_ID);
		// District
		fields[lpos++] = genfield(threadP,11,11, 4,&data->in.s_C_ID);
		// Customer #
		fields[lpos++] = genfield(threadP,29,12,16,(char *)data->in.s_C_LAST);   // Name
		fields[lpos++] = genfield(threadP,33,11, 5,&data->in.s_W_ID);
		// Cust-Warehouse
		fields[lpos++] = genfield(threadP,54,11, 2,&data->in.s_D_ID);
		// Cust-District
		fields[lpos++] = genfield(threadP,23,17, 8,&data->in.s_H_AMOUNT);
// Amount Paid
		fields[lpos++] = NULL;

		return;
}

/*
*****************************************************************
** Name				: Payment::validate
** Description:
**							Validate entries
** Parameters :
**							None
** Returns		:
**							int - Return code
** Comments		:
**
*****************************************************************
*/

int Payment::validate()
{
		// validate district
		if(!fields[0]->pos)
		{
				pos = 0;
				message(threadP,"District ID is a required field");
				return 0;
		}

		// customer id, if zero we need to collect name

		if(!fields[1]->pos && !fields[2]->pos)
		{
				if(data->inputName < 2)
				{
						data->inputName++; pos = 1;
						message(threadP, "Customer ID or Name is required");
						return 0;
				}
		}

		if(!fields[2]->pos && data->inputName >= 2)
		{
				if(data->inputName < 4)
				{
						data->inputName++; pos = 2;
						message(threadP, "Customer ID or Name is required");
						return 0;
				}
				else
				{
						data->inputName = 1; pos = 1;
						message(threadP, "Customer ID or Name is required");
						return 0;
				}
		}
		data->inputName = 0;

		// Customer warehouse
		if(!fields[3]->pos)
		{
				pos = 3;
```

```
                message(threadP, "Customer Warehouse is a required field");
                return 0;
        }

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

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

        data->in.s_W_ID = user_dataP->warehouse;
        return 1;
}

/*
*****************************************************************
** Name                         : Payment::respond
** Description:
**                                                      Display values
** Parameters :
**                                                      None
** Returns              :
**                                                      int - Return code
** Comments             :
**
*****************************************************************
*/

int Payment::respond()
{
        display_status(data->header.sql_code);
        // check transaction return code
        if(data->header.sql_code != OK)
        {
                return ERR;
        }

        char buf[32];
        position(threadP,52,6);                 format_int(buf,3,data->in.s_D_ID);
                                                        threadP->write(buf,2);
        position(threadP,33,11);        format_int(buf,6,data->in.s_C_W_ID);
                                        threadP->write(buf,4);
        position(threadP,54,11);        format_int(buf,3,data->in.s_C_D_ID);
                                        threadP->write(buf,2);
        position(threadP,7, 4);                 format_date(buf,11,data->in.s_H_DATE_time);
                                        threadP->write(buf,10);
        position(threadP, 1,7);                 threadP->write(data->out.s_W_STREET_1);
        position(threadP,42,7);                 threadP->write(data->out.s_D_STREET_1);
        position(threadP, 1,8);                 threadP->write(data->out.s_W_STREET_2);
        position(threadP,42,8);                 threadP->write(data->out.s_D_STREET_2);
        position(threadP, 1,9);                 threadP->write(data->out.s_W_CITY);
        position(threadP,22,9);                 threadP->write(data->out.s_W_STATE);
        position(threadP,25,9);                 format_zip(buf,10,data->out.s_W_ZIP);
                                                threadP->write(buf,10);
        position(threadP,42,9);                 threadP->write(data->out.s_D_CITY);
        position(threadP,63,9);                 threadP->write(data->out.s_D_STATE);
        position(threadP,66,9);                 format_zip(buf,10,data->out.s_D_ZIP);
                                                threadP->write(buf,10);
        position(threadP,11,11);        format_int(buf,6,data->out.s_C_ID);
                                                threadP->write(buf,4);
        position(threadP, 9,12);        threadP->write(data->out.s_C_FIRST);
        position(threadP,26,12);        threadP->write(data->out.s_C_MIDDLE);
        position(threadP,29,12);        threadP->write(data->out.s_C_LAST);
        position(threadP,58,12);        format_datetime(buf,20,data->out.s_C_SINCE_time);
threadP->write(buf,19);
        position(threadP, 9,13);        threadP->write(data->out.s_C_STREET_1);
        position(threadP,58,13);        threadP->write(data->out.s_C_CREDIT);
        position(threadP, 9,14);        threadP->write(data->out.s_C_STREET_2);
        position(threadP,58,14);
format_float(buf,6,2,(double)(data->out.s_C_DISCOUNT/100.0));           threadP->write(buf,6);
        position(threadP, 9,15);        threadP->write(data->out.s_C_CITY);
        position(threadP,30,15);        threadP->write(data->out.s_C_STATE);
        position(threadP,33,15);        format_zip(buf,10,data->out.s_C_ZIP);
                                        threadP->write(buf,10);
        position(threadP,58,15);        format_phone(buf,18,data->out.s_C_PHONE);
                                        threadP->write(buf,18);
        position(threadP,17,17);
format_money(buf,15,(double)(data->in.s_H_AMOUNT/100.0));
        threadP->write(buf,14);
        position(threadP,55,17);
format_money(buf,16,(double)(data->out.s_C_BALANCE/100.0));
threadP->write(buf,15);
        position(threadP,17,18);
format_money(buf,15,(double)(data->out.s_C_CREDIT_LIM/100.0));
threadP->write(buf,14);

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

                        size -= 50;
                        if(size <= 0) break;
                }
        }

        return 0;
}

/*
*****************************************************************
** Name                         : Payment::process
** Description:
**                                                      Run transaction
** Parameters :
**                                                      None
** Returns              :
**                                                      int - return code
** Comments             :
**
*****************************************************************
*/

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

        data = send_payment(threadP->contextP,data);
        update_address((void *) data);

        return 0;
}

////////////////////////////////////////////////////////////////////
// OrderStatus class
////////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name                         : OrderStatus::OrderStatus
** Description:
**                                                      Constructor
** Parameters :
**                                                      User_data * - User
**                                                      Thread_data * - Thread
** Returns              :
**                                                      None
** Comments             :
**
*****************************************************************
*/

OrderStatus::OrderStatus(User_data *udP,Thread_data *threadP) : Screen(udP,threadP)
{
        data                                                    = (orderStatus_data_t *)
tuxBufferAllocate(sizeof(orderStatus_data_t));

        tran_type               = ORDER_STATUS_TRANS;
        data_len                                = sizeof(orderStatus_data_t);
        dataptr                                 = data;

        status_x                                = 1;
        status_y                                = 25;

        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 pos = 0; fields = new Field *[4];
        fields[pos++] = genfield(threadP,29,4, 2,&data->in.s_D_ID);
                                // District */
        fields[pos++] = genfield(threadP,11,5, 4,&data->in.s_C_ID);
                                // Customer ID */
        fields[pos++] = genfield(threadP,44,5,16,(char *) data->in.s_C_LAST);                   //
Customer Name */
        fields[pos++] = NULL;

        return;
};

/*
*****************************************************************
** Name                         : OrderStatus::validate
```

```
** Description:
**                                                              Validate field entries
** Parameters :
**                                                              None
** Returns              :
**                                                              int - return code
** Comments             :
**
*****************************************************************
*/

int OrderStatus::validate()
{
        // validate district
        if(!fields[0]->pos)
        {
                pos = 0;
                message(threadP,"District ID is a required field");
                return 0;
        }

        // customer id, if zero we need to collect name

        if(!fields[1]->pos && !fields[2]->pos)
        {
                if(data->inputName < 2)
                {
                        data->inputName++; pos = 1;
                        message(threadP, "Customer ID or Name is required");
                        return 0;
                }
        }

        if(!fields[2]->pos && data->inputName >= 2)
        {
                if(data->inputName < 4)
                {
                        data->inputName++; pos = 2;
                        message(threadP, "Customer ID or Name is required");
                        return 0;
                }
                else
                {
                        data->inputName = 1; pos = 1;
                        message(threadP, "Customer ID or Name is required");
                        return 0;
                }
        }
        data->inputName = 0;

        data->in.s_W_ID = user_dataP->warehouse;
        return 1;
}

/*
*****************************************************************
** Name                         : OrderStatus::respond
** Description:
**                                                              Responce back to user
** Parameters :
**                                                              None
** Returns              :
**                                                              int - return code
** Comments             :
**
*****************************************************************
*/

int OrderStatus::respond()
{
        // check transaction return code
        display_status(data->header.sql_code);
        if (data->header.sql_code != OK)
                return ERR;

        char buf[32];
        position(threadP,11,5);         format_int(buf,6,data->out.s_C_ID);
                                        threadP->write(buf,4);
        position(threadP,24,5);         threadP->write(data->out.s_C_FIRST);
        position(threadP,41,5);         threadP->write(data->out.s_C_MIDDLE);
        position(threadP,44,5);         threadP->write(data->out.s_C_LAST);
        position(threadP,15,6);
format_money(buf,11,(double)(data->out.s_C_BALANCE/100.0));
threadP->write(buf,10);
        position(threadP,15,8);         format_int(buf,10,data->out.s_O_ID);
                threadP->write(buf,8);
        position(threadP,38,8);
format_datetime(buf,20,data->out.s_O_ENTRY_D_time);threadP->write(buf,19);

        if(data->out.s_O_CARRIER_ID > 0)
        {
                position(threadP,76,8);
                format_int(buf,3,data->out.s_O_CARRIER_ID);
                threadP->write(buf,2);
        }

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

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

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

                position (threadP,32,i + 10);
                format_money(buf,10,(double)(data->out.item[i].s_OL_AMOUNT/100.0));
                threadP->write(buf,9);

                position (threadP,47,i + 10);
                format_date(buf,11,data->out.item[i].s_OL_DELIVERY_D_time);
                threadP->write(buf,10);
        }

        return 0;
}

/*
*****************************************************************
** Name                         : OrderStatus::process
** Description:
**                                                              Run transaction
** Parameters :
**                                                              None
** Returns              :
**                                                              int - return code
** Comments             :
**
*****************************************************************
*/

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

        data = send_order_status(threadP->contextP,data);
        update_address((void *) data);

        return 0;
}

////////////////////////////////////////////////////////////////////
// Delivery class
////////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name                         : Delivery::Delivery
** Description:
**                                                              Constructor
** Parameters :
**                                                              User_data * - User
**                                                              Thread_data * - Thread
** Returns              :
**                                                              None
** Comments             :
**
*****************************************************************
*/

Delivery::Delivery(User_data *udP,Thread_data *threadP) : Screen(udP,threadP)
{
        data                                                    = (delivery_data_t *)
tuxBufferAllocate(sizeof(delivery_data_t));

        tran_type               = DELIVERY_TRANS;
        data_len                                = sizeof(delivery_data_t);
        dataptr                                 = data;

        status_x                                = 1;
        status_y                                = 8;

        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 pos = 0; fields = new Field *[2];
        fields[pos++] = genfield(threadP,17,6,2,&data->in.s_O_CARRIER_ID);   // Carrier Number
        fields[pos++] = NULL;

        return;
```

```
}

/*
*****************************************************************
** Name                         : Delivery::validate
** Description:
**                                                      Validate field entries
** Parameters :
**                                                      None
** Returns              :
**                                                      int - return code
** Comments             :
**
*****************************************************************
*/

int Delivery::validate()
{
        // validate carrier
        if(!fields[0]->pos)
        {
                pos = 0;
                message(threadP,  "Carrier ID is a required field");
                return 0;
        }

        time((time_t *) &(data->in.s_O_DELIVERY_D_time));
        data->in.s_W_ID = user_dataP->warehouse;
        return 1;
}

/*
*****************************************************************
** Name                         : Delivery::respond
** Description:
**                                                      Responce back to user
** Parameters :
**                                                      None
** Returns              :
**                                                      int - return code
** Comments             :
**
*****************************************************************
*/

int Delivery::respond()
{
        // check transaction return code
        if(data->header.sql_code == OK)
        {
                position(threadP,status_x,status_y);
                threadP->write("Execution Status: Delivery has been queued");
        }
        else
        {
                display_status(data->header.sql_code);
                return ERR;
        }

        return OK;
}

/*
*****************************************************************
** Name                         : Delivery::process
** Description:
**                                                      Run transaction
** Parameters :
**                                                      None
** Returns              :
**                                                      int - return code
** Comments             :
**
*****************************************************************
*/

int Delivery::process()
{
        if(tran_type != DELIVERY_TRANS)
                return 0;

        data = send_delivery(threadP->contextP,data);
        update_address((void *) data);

        return 0;
}

////////////////////////////////////////////////////////////////////////
// StockLevel class
////////////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name                         : StockLevel::StockLevel
** Description:
**                                                      Constructor
** Parameters :
**                                                      User_data * - User
**                                                      Thread_data * - Thread
** Returns              :
**                                                      None
** Comments             :
**
*****************************************************************
*/

StockLevel::StockLevel(User_data *udP,Thread_data *threadP) : Screen(udP,threadP)
{
        data                                    = (stockLevel_data_t *)
tuxBufferAllocate(sizeof(stockLevel_data_t));

        tran_type               = STOCK_TRANS;
        data_len                        = sizeof(stockLevel_data_t);
        dataptr                         = data;

        status_x                        = 1;
        status_y                        = 10;

        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;

        int pos = 0; fields = new Field *[2];
        fields[pos++] = genfield(threadP,24,6,2,&data->in.s_threshold);          // Threshold
        fields[pos++] = NULL;

        return;
}

/*
*****************************************************************
** Name                         : StockLevel::validate
** Description:
**                                                      Validate field entries
** Parameters :
**                                                      None
** Returns              :
**                                                      int - return code
** Comments             :
**
*****************************************************************
*/

int StockLevel::validate()
{
        // validate stock level
        if(data->in.s_threshold <= 0)
        {
                pos = 0;
                message(threadP,"A positive non-zero threshold is required");
                return 0;
        }

        data->in.s_W_ID = user_dataP->warehouse; // Waarehouse
        data->in.s_D_ID = user_dataP->district;       // District
        return 1;
}

/*
*****************************************************************
** Name                         : StockLevel::respond
** Description:
**                                                      Responce back to user
** Parameters :
**                                                      None
** Returns              :
**                                                      int - return code
** Comments             :
**
*****************************************************************
*/

int StockLevel::respond()
{
         // check trasaction status
        display_status(data->header.sql_code);
        if(data->header.sql_code != OK)
                return ERR;

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

        return 0;
```

```
}

/*
*****************************************************************
** Name                                  : StockLevel::process
** Description:
**                                                               Run transaction
** Parameters :
**                                                               None
** Returns                 :
**                                                               int - return code
** Comments                :
**
*****************************************************************
*/

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

        data = send_stock_level(threadP->contextP,data);
        update_address((void *) data);

        return 0;
}

//////////////////////////////////////////////////////////////////
// Login class
//////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name                                  : Login::Login
** Description:
**                                                               Constructor
** Parameters :
**                                                               User_data * - User
**                                                               Thread_data * - Thread
** Returns                 :
**                                                               None
** Comments                :
**
*****************************************************************
*/

Login::Login(User_data *udP,Thread_data *threadP) : Screen(udP,threadP)
{
        int pos                                         = 0;
        tran_type                       = NULL;
        status_x                                        = 1;
        status_y                                        = 24;

        dataptr                                         = NULL;
        data_len                                        = 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;

        return;
}

/*
*****************************************************************
** Name                                  : Login::validate
** Description:
**                                                               Validate field entries
** Parameters :
**                                                               None
** Returns                 :
**                                                               int - return code
** Comments                :
**
*****************************************************************
*/

int Login::validate()
{
        // validate warehouse
        if(!fields[0]->pos)
        {
                pos = 0;
                message(threadP,"Warehouse ID is a required field");
                return 0;
        }

        // validate district
        if(!fields[1]->pos)
        {
                pos = 1;
                message(threadP,"District ID is a required field");
                return 0;
        }

        return 1;
}

//////////////////////////////////////////////////////////////////
// Menu class
//////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name                                  : Menu::Menu
** Description:
**                                                               Constructor
** Parameters :
**                                                               User_data * - User
**                                                               Thread_data * - Thread
** Returns                 :
**                                                               None
** Comments                :
**
*****************************************************************
*/

Menu::Menu(User_data *udP,Thread_data *threadP) : Screen(udP,threadP)
{
        int pos                                         = 0;
        tran_type                       = NULL;
        status_x                                        = 1;
        status_y                                        = 24;

        screen                                          = static_screen;
        screen_len                      = static_screen_len;
        empty_fields            = NULL;
        empty_fields_len        = 0;

        fields                                          = NULL;
        return;
}

//////////////////////////////////////////////////////////////////
```

tuxClientScreen.hpp

```
/*
*****************************************************************
** Project                 : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                                  : tuxClientScreen.hpp
** Title                                 : Tux Screen interface
*****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                              :
**
*****************************************************************
*/

#ifndef __tuxClientScreen_hpp__
#define __tuxClientScreen_hpp__

#include <commonTuxDB2.hpp>
#include <tuxClientCommon.hpp>

#include <tuxClientTerminal.hpp>
#include <tuxClientField.hpp>
#include <tuxClientMonitor.hpp>
#include <tuxClientInf.hpp>

//////////////////////////////////////////////////////////////////
// Classes
//////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////
// User
//////////////////////////////////////////////////////////////////

class User_data
{
        // attributes
        public:
                int warehouse;
                int district;
};

//////////////////////////////////////////////////////////////////
// Thread
//////////////////////////////////////////////////////////////////
```

```
class Thread_data : public Terminal
{
        // constructors
        public:
                Thread_data(int infd,int outfd,void *conP) :
Terminal(infd,outfd),contextP(conP) {return;}

        // attributes
        public:
                void            *contextP;
};

////////////////////////////////////////////////////////////////
// Screen
////////////////////////////////////////////////////////////////

class Screen
{
        // constructors / destructors
        public:
                inline          Screen(User_data *udP,Thread_data *thrP);
                virtual         ~Screen();

        // class functions
        public:
                virtual int                                             reset();
                virtual int                                             present();
                virtual int
present_empty_fields();
                virtual int                                             process();
                virtual int                                             user_input();

                virtual char                                    *isa()
                                {return "Screen";}
                virtual int                                             validate()
                                {return 1;}
                virtual int                                             respond()
                                {return 0;}

                int
handle();
                int
display_status(int status);

                void
update_address(void *bufferAddress);

        // attributes
        protected:
                static char end_str[];

                static int                                      end_str_len;

                int
has_data;
                int
screen_len;
                int
session_data_len;
                int
empty_fields_len;
                int
pos;
                int
status_x, status_y;
                int
data_len;

                int
tran_type;
                char
*session_data;

                char
*screen;
                char
*empty_fields;

                void
*dataptr;
                Thread_data                             *threadP;

        public:
                User_data                               *user_dataP;
                Field
**fields;
};

////////////////////////////////////////////////////////////////
// Login
////////////////////////////////////////////////////////////////

class Login : public Screen
{
        // constructors
        public:
                Login(User_data *udP,Thread_data *thrP);

        // functions
        public:
                int             validate();

        // attributes
        protected:
                static char                     static_screen[];
                static char                     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;
};

////////////////////////////////////////////////////////////////
// NerwOrder
////////////////////////////////////////////////////////////////

class NewOrder : public Screen
{
        // costructors
        public:
                NewOrder(User_data *udP,Thread_data *thrP);

        // functions
        protected:
                void            swap_fields(int i,int j);

        public:
                int             reset();
                int             validate();
                int             process();
                int             respond();

        // attributes
        protected:
                static char                     static_screen[];
                static char                     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;

        public:
                newOrder_data_t                 *data;
};

////////////////////////////////////////////////////////////////
// Payment
////////////////////////////////////////////////////////////////

class Payment : public Screen
{
        // constructors
        public:
                Payment(User_data *udP,Thread_data *thrP);

        // functions
        public:
                int             validate();
                int             process();
                int             respond();

        // Attributes
        protected:
                static char                     static_screen[];
                static char                     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_t                                  *data;
};

////////////////////////////////////////////////////////////////
// Order Status
////////////////////////////////////////////////////////////////

class OrderStatus : public Screen
{
        // constructors
        public:
                OrderStatus(User_data *udP,Thread_data *thrP);

        // functions
        public:
                int             validate();
                int             process();
                int             respond();
```

```
        // attributes
        protected:
                static char                             static_screen[];
                static char                             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_t      *data;
};

//////////////////////////////////////////////////////////////////
// Delivery
//////////////////////////////////////////////////////////////////

class Delivery : public Screen
{
        // costructors
        public:
                Delivery(User_data *udP,Thread_data *thrP);

        // functions
        public:
                int             validate();
                int             process();
                int             respond();

        // attributes
        protected:
                static char             static_screen[];
                static char             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_t         *data;
};

//////////////////////////////////////////////////////////////////
// Stock Level
//////////////////////////////////////////////////////////////////

class StockLevel : public Screen
{
        // constructors
        public:
                StockLevel(User_data *udP,Thread_data *thrP);

        // functions
        public:
                int             validate();
                int             process();
                int             respond();

        // attributes
        protected:
                static char             static_screen[];
                static char             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_t       *data;
};

//////////////////////////////////////////////////////////////////
// Menu
//////////////////////////////////////////////////////////////////

class Menu : public Screen
{
        // constructors
        public:
                Menu(User_data *udP,Thread_data *thrP);

        // attributes
        protected:
                static char             static_screen[];
                static char             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;
};

//////////////////////////////////////////////////////////////////
// inlines
//////////////////////////////////////////////////////////////////

#include <tuxClientScreen.inl>
#endif
//////////////////////////////////////////////////////////////////
```

tuxClientScreen.inl

```
/*
****************************************************************
** Project                      : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                         : tuxClientScreen.inl
** Title                        : Tux Screen interface inline class functions
****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                     :
**
****************************************************************
*/

#ifndef __tuxClientScreen_inl__
#define __tuxClientScreen_inl__

//////////////////////////////////////////////////////////////////
// screen class
//////////////////////////////////////////////////////////////////

/*
****************************************************************
** Name                         : Screen::Screen
** Description:
**                                                              Screen constructor
** Parameters :
**                                                              User_data * - user
**                                                              Thread_data * - thread
** Returns                      :
**                                                              None
** Comments                     :
**
****************************************************************
*/

inline Screen::Screen(User_data *udP,Thread_data *thrP)
{
        user_dataP                              = udP;
        threadP                                         = thrP;

        has_data                                        = 0;
        pos                                                     = 0;
        fields                                          = NULL;

        screen                                          = NULL;
        empty_fields            = NULL;
        session_data            = NULL;

        screen_len                              = 0;
        session_data_len        = 0;
        empty_fields_len        = 0;

        return;
}
#endif
//////////////////////////////////////////////////////////////////
```

tuxClientTerminal.cpp

```
/*
****************************************************************
** Project                      : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                         : tuxClientTerminal.cpp
** Title                        : Tux client terminal
****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
```

```
**
** History      :
**             Developed at IBM Austin by the AIX RS/6000
**             performance group.
**
** Comments                                  :
**
*****************************************************************
*/

#include <tuxClientTerminal.hpp>

//////////////////////////////////////////////////////////////////////
// Externals
//////////////////////////////////////////////////////////////////////

char const * const blanks                          = "                    ";
char const * const underscores                = "________________________________";
char const * const backspaces                 = "\b\b\b\b\b\b\b\b\b\b";

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

/*
*****************************************************************
** Name                                     : Terminal::write
** Description:
**                                                                Write
** Parameters :
**                                                                const void * - buffer
**                                                                size_t - size
** Returns                   :
**                                                                None
** Comments                  :
**
*****************************************************************
*/

void Terminal::write(const void *buf,size_t size)
{
        char            logmsg[MAX_MSG_LEN];

        if (IOError) return;

        sprintf(logmsg,"write('%*.*s', %d);\n",size,size,buf,size); err_printf(logmsg);
        output.queue(buf,size);
        #if COMM_LOG
                write_comms_entry((char *) buf,'O');
        #endif

        // Don't write anything until there is no input
        if(!Hold && input.len() == 0)
                flush();

        return;
}

/*
*****************************************************************
** Name                                     : Terminal::write
** Description:
**                                                                Write
** Parameters :
**                                                                const void * - buffer
** Returns                   :
**                                                                None
** Comments                  :
**
*****************************************************************
*/

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

/*
*****************************************************************
** Name                                     : Terminal::read
** Description:
**                                                                Read
** Parameters :
**                                                                const void * - buffer
**                                                                size_t - size
** Returns                   :
**                                                                ssize_t - length
** Comments                  :
**
*****************************************************************
*/

ssize_t Terminal::read(void *buf,size_t size)
{
        int             rc;
        char            logmsg[MAX_MSG_LEN];
        int             inputLen=input.len();

        if (IOError) return 0;
#if COMM_LOG
        sprintf(logmsg,"input.len():%d  size:%d\n",input.len(),size);
        err_printf(logmsg);

        sprintf(logmsg,"input.free():%d \n",input.free());
        err_printf(logmsg);
#endif

        while(input.len() < size)
        {
                rc = ::read(in_fd,input.ptr(),input.free());
                sprintf(logmsg,"::read('%*.*s', %d) = %d;\n",rc,rc,input.ptr(),input.free(),rc);
                err_printf(logmsg);

                #if COMM_LOG
                        inputLen=input.len();

                        sprintf(logmsg,"input.len():%d  size:%d\n",input.len(),size);
                err_printf(logmsg);

                        sprintf(logmsg,"input.free():%d \n",input.free());
        err_printf(logmsg);

                        write_comms_entry((char *) input.ptr(),'I');
                #endif

                if(inlog)
                {
                        fwrite(input.ptr(),rc,1,inlog);
                        fflush(inlog);
                }

                if(rc > 0)
                {
                        input.queue(rc);
                }
                else
                {
                        if (rc <= 0)
                        {
                                IOError = 1;
                                return 0;
                        }
                }
        }
        #if COMM_LOG
                inputLen=input.len();

        sprintf(logmsg,"input.len():%d  size:%d\n",input.len(),size);
        err_printf(logmsg);
        #endif

        memcpy(buf,input.ptr(),size);
        input.dequeue(size);

        sprintf(logmsg,"read('%*.*s', %d) = %d;\n", size, size, buf, size, size); err_printf(logmsg);
        return size;
}

/*
*****************************************************************
** Name                                     : Terminal::flush
** Description:
**                                                                Flush
** Parameters :
**                                                                None
** Returns                   :
**                                                                None
** Comments                  :
**
*****************************************************************
*/

void Terminal::flush()
{
        char            logmsg[MAX_MSG_LEN];
        sprintf(logmsg,"flush();\n"); err_printf(logmsg);

        Hold = 0;
        if (IOError) return;

        while(output.len())
        {
                sprintf(logmsg,"::write('%*.*s',
%d);\n",output.len(),output.len(),output.ptr(),output.len());
                err_printf(logmsg);

                int rc = ::write(out_fd, output.ptr(), output.len());

                if(outlog)
                {
                        fwrite(output.ptr(),rc,1,outlog);
                        fflush(outlog);
                }

                if(rc > 0)
                {
                        output.dequeue(rc);
                }
```

```
                else
                {
                        if (rc < 0)
                        {
                                sprintf(logmsg,"Error writing data!\n");
err_printf(logmsg);
                                IOError = 1; return;
                        }
                }
        }

        return;
}

/*
*****************************************************************
** Name                         : Terminal::Terminal
** Description:
**                                                      Constructor
** Parameters :
**                                                      int - input descriptor
**                                                      int - output descriptor
** Returns              :
**                                                      None
** Comments             :
**
*****************************************************************
*/

Terminal::Terminal(int in, int out) : input(2048), output(4096)
{
        int                                             rc;
        char                                            logmsg[MAX_MSG_LEN];
        struct termios  buf;

        Hold                    = 0;
        IOError                 = 0;

        debugfile       = NULL;
        inlog                   = NULL;
        outlog                  = NULL;

        // store file descriptors
        in_fd = in;
        if(out < 0)
                out_fd = in;
        else out_fd = out;

        if((rc = tcgetattr(in_fd,&save_term)) < 0)
                return;

        // set attributes
        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;

        sprintf(logmsg,"echo off - tcsetattr on %d\n",in_fd);
        err_printf(logmsg);

        if (tcsetattr(in_fd,TCSAFLUSH, &buf) < 0)
                return;

        return;
}

/*
*****************************************************************
** Name                         : Terminal::~Terminal
** Description:
**                                                      Destructor
** Parameters :
**                                                      None
** Returns              :
**                                                      None
** Comments             :
**
*****************************************************************
*/

Terminal::~Terminal()
{
        if (tcsetattr(in_fd,TCSAFLUSH,&save_term) < 0)
                return;

        return;
}

////////////////////////////////////////////////////////////////////////
// Utilities to help manage terminal interaction
////////////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name                         : position
** Description:
**                                                      Set cursor position
** Parameters :
**                                                      int - x pos
**                                                      int - y pos
**                                                      char * - buffer
** Returns              :
**                                                      Return code
** Comments             :
**
*****************************************************************
*/

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;
}

/*
*****************************************************************
** Name                         : position
** Description:
**                                                      Set cursor position
** Parameters :
**                                                      Terminal * - output device
**                                                      int - x pos
**                                                      int - y pos
** Returns              :
**                                                      Return code
** Comments             :
**
*****************************************************************
*/

int position(Terminal *terminal,int x,int y)
{
        char buf[16];

        position(x,y,buf);
        terminal->write(buf);

        return 0;
}

/*
*****************************************************************
** Name                         : position
** Description:
**                                                      Write EOS
** Parameters :
**                                                      Terminal * - output device
** Returns              :
**                                                      Return code
** Comments             :
**
*****************************************************************
*/

int clear_eos(Terminal *terminal)
{
        terminal->write(ESC "[J");
        return 0;
}

/*
*****************************************************************
** Name                         : position
** Description:
**                                                      Write EOS
** Parameters :
**                                                      char * - buffer
** Returns              :
**                                                      Return code
** Comments             :
**
*****************************************************************
*/

int clear_eos(char *buf)
{
        buf[0] = ESCc;
        buf[1] = '[';
```

```
		buf[2] = 'J';

		return 0;
}

/*
*****************************************************************
** Name				: position
** Description:
**					Write EOS
** Parameters :
**					Terminal * - output device
**					char * - text
**					int - flush flag
** Returns		:
**					Return code
** Comments		:
**
*****************************************************************
*/

int message(Terminal *terminal,char *text,int need_flush)
{
		position(terminal,1,25);

		terminal->write(text);
		clear_eos(terminal);

		if(need_flush)
			terminal->flush();

		return 0;
}

/////////////////////////////////////////////////////////////////////
```

tuxClientTerminal.hpp

```
/*
*****************************************************************
** Project		: AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name				: tuxClientTerminal.hpp
** Title				: Terminal control
*****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History	:
**		Developed at IBM Austin by the AIX RS/6000
**		performance group.
**
** Comments		:
**
*****************************************************************
*/

#ifndef __tuxClientTerminal_hpp__
#define __tuxClientTerminal_hpp__

#include <commonTuxDB2.hpp>
#include <tuxClientCommon.hpp>
#include <tuxClientMonitor.hpp>

/////////////////////////////////////////////////////////////////////
// Definitions (vt100)
/////////////////////////////////////////////////////////////////////

#define ESC "\033"
#define ESCc '\033'

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

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

/////////////////////////////////////////////////////////////////////
// Enumeration
/////////////////////////////////////////////////////////////////////

typedef enum
{
		liberty_term,
		curses_term,
		emulator_term
} term_type_t;

/////////////////////////////////////////////////////////////////////
// Classes
/////////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////////
// TerminalBuffer
/////////////////////////////////////////////////////////////////////

class TerminalBuffer
{
		// Freinds
		friend class Terminal;

		// constructor
		private:
			inline TerminalBuffer(int size);

		// functions
		private:
			inline void	mark();
			inline void	pop();
			inline void queue(const void *buf,int size);
			inline void dequeue(int size);

			void		unmark()
{if(num_marks <= 0) return;num_marks--;return;}
			void		queue(int size) {Pos += size;return;}

			int		pos()
{return Pos;}
			int		start()
{return Start;}
			int		len()
{return Pos - Start;}
			int		free()
{return BufSize - Pos - 1;}
			int		lastmark()				{return
(num_marks) ? marks[num_marks - 1] : 999;}

			void		pos(int P)				{Pos = P;}
			void		start(int S)		{Start = S;}
			void		*ptr()
{return &buffer[Start];}

		// attributes
		private:
			enum		{NUMMARKS = 8};
			int		marks[NUMMARKS];

			char		*buffer;
			int		BufSize;

			int		Pos;
			int		Start;
			int		num_marks;
};

/////////////////////////////////////////////////////////////////////
// Terminal
/////////////////////////////////////////////////////////////////////

class Terminal
{

		// Constructors & Destructors
		public:
			Terminal(int in = 0,int out = 1);
			~Terminal();

		// Functions
		public:
			ssize_t read(void *buf,size_t size);

			void flush();
			void write(const void *buf);
			void write(const void *buf,size_t size);

			void mark()		{output.mark();return;}
			void unmark() {output.unmark();return;}
			void pop()		{output.pop();return;}
			void hold()		{Hold = 1;return;}


		// Attributes
		private:
			int
in_fd;
			int
out_fd;
			int
Hold;

			struct termios		save_term;

			TerminalBuffer				input;
			TerminalBuffer				output;

			FILE
*debugfile;
			FILE
*inlog;
			FILE
*outlog;

		public:
```

```
                                int
IOError;
};

////////////////////////////////////////////////////////////////////
// Terminal externs
////////////////////////////////////////////////////////////////////

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

////////////////////////////////////////////////////////////////////
// Functions
////////////////////////////////////////////////////////////////////

int position(int x,int y,char *buf);
int position(Terminal *threadP,int x,int y);
int clear_eos(Terminal *threadP);
int message(Terminal *threadP,char *text,int need_flush = 1);
int clear_eos(char *buf);

#include <tuxClientTerminal.inl>
#endif
////////////////////////////////////////////////////////////////////
```

tuxClientTerminal.inl

```
/*
*********************************************************************
** Project                      : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                         : tuxClientTerminal.inl
** Title                        : Tux terminal control inline class functions
*********************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                     :
**
*********************************************************************
*/

#ifndef __tuxClientTerminal_inl__
#define __tuxClientTerminal_inl__

////////////////////////////////////////////////////////////////////
// TerminalBuffer class
////////////////////////////////////////////////////////////////////

/*
*********************************************************************
** Name                         : TerminalBuffer::TerminalBuffer
** Description:
**                                                      TerminalBuffer constructor
** Parameters :
**                                                      int - buffer size
** Returns              :
**                                                      None
** Comments             :
**
*********************************************************************
*/

inline TerminalBuffer::TerminalBuffer(int size)
{
        BufSize         = size;
        buffer          = new char [BufSize];

        Pos                             = 0;
        Start                           = 0;
        num_marks       = 0;

        return;
}

/*
*********************************************************************
** Name                         : TerminalBuffer::mark
** Description:
**                                                      Add mark
** Parameters :
**                                                      None
** Returns              :
**                                                      None
** Comments             :
**
*********************************************************************
*/

inline void TerminalBuffer::mark()
{
        if(num_marks < NUMMARKS)
        {
                marks[num_marks++] = Pos;
        }
        else
        {
                fprintf(stderr, "Buffer mark overflow\n");
                exit_program(1);
        }

        return;
}

/*
*********************************************************************
** Name                         : TerminalBuffer::pop
** Description:
**                                                      Pop off a mark
** Parameters :
**                                                      None
** Returns              :
**                                                      None
** Comments             :
**
*********************************************************************
*/

inline void TerminalBuffer::pop()
{
        if(num_marks <= 0)
                return;

        if(marks[num_marks-1] >= Start)
                Pos = marks[--num_marks];
        else num_marks=0;

        return;
}

/*
*********************************************************************
** Name                         : TerminalBuffer::queue
** Description:
**                                                      Queue terminal data
** Parameters :
**                                                      void * - data buffer
**                                                      int size
** Returns              :
**                                                      None
** Comments             :
**
*********************************************************************
*/

inline void TerminalBuffer::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_program(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_program(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;
        }

        // copy
        memcpy(&buffer[Pos],buf,size);
        Pos += size;

        return;
}

/*
*********************************************************************
```

```
** Name                                  : TerminalBuffer::dequeue
** Description:
**                                                                dequeue terminal data
** Parameters :
**                                                                int size
** Returns                 :
**                                                                None
** Comments                :
**
*****************************************************************
*/

inline void TerminalBuffer::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;
        }

        return;
}

#endif
//////////////////////////////////////////////////////////////////////////
```

tuxClientThread.cpp

```
/*
*****************************************************************
** Project                 : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                                  : tuxClientThread.cpp
** Title                                 : Client thread communicating with RTE
*****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History       :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                              :
**
*****************************************************************
*/

#include <tuxClientThread.hpp>

/*
*****************************************************************
** Name                                  : client_last_time
** Description:
**                                                                returns a pointer to the structure
in the thread
**                                                                data that contains the time
corresponding to the
**                                                                threads current state
** Parameters :
**                                                                thread_descr_t - thread
** Returns                 :
**                                                                timeval * - thread last time
** Comments                :
**
*****************************************************************
*/

struct timeval *client_last_time(thread_descr_t *descrP)
{
        char                                                    logmsg[MAX_MSG_LEN];
        struct timeval *lastTimeP = &descrP->done;

        switch(descrP->state)
        {
        case thread_state_init:
                // Thread is initializing - no trans 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:
                sprintf(logmsg,"client_last_time: bad state: %d\n",descrP->state);
err_printf(logmsg);
                lastTimeP = &descrP->done;
                break;
        }

        return(lastTimeP);
}

/*
*****************************************************************
** Name                                  : set_client_debug_state
** Description:
**                                                                Set debug
** Parameters :
**                                                                void * - thread contex
**                                                                int - state
**                                                                int - transaction
** Returns                 :
**                                                                None
** Comments                :
**
*****************************************************************
*/

void set_client_debug_state(void *contextP,int state,int tran)
{
        struct timezone                 tz;

        thread_info_t                   *thread_context                 = (thread_info_t *) contextP;
        thread_descr_t                  *descrP                                                 =
&thread_context->descr;

        descrP->state = (thread_state_t) state;

        gettimeofday(client_last_time(descrP),&tz);

        if(state == thread_state_called)
                descrP->tran = tran;

        return;
}

/*
*****************************************************************
** Name                                  : thread_state_to_str
** Description:
**                                                                Decode thread state to report
** Parameters :
**                                                                int - state
** Returns                 :
**                                                                Thread state
** Comments                :
**                                                                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.
**
*****************************************************************
*/

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;
		default:
			ret_val = "-Unknown-"; break;
		}

		return(ret_val);
}

/*
****************************************************************
** Name				: print_rt_avg
** Description:
**                                                              Print runtime avg
** Parameters :
**                                                              total_tran_count_t * - current
**                                                              total_tran_count_t * - previous
**                                                              int - transaction
** Returns		:
**                                                              None
** Comments		:
**
****************************************************************
*/

void print_rt_avg(total_tran_count_t *curP,total_tran_count_t *prevP,int type)
{
		char				logmsg[MAX_MSG_LEN];
		static char	*names[]	= {"0", "no", "pa", "os", "dl", "sl"};

		sprintf(logmsg,"%s RT avg: ",type ? "server" : "client");
		err_printf(logmsg);

		for (int 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];

			if(num_trans > 0)
				fprintf(ERROUT," %s %.0f,",names[i],(double) (rt_diff) /
(num_trans));
		}

		fprintf(ERROUT,"\n");
		return;
}

/*
****************************************************************
** Name				: bg_thread
** Description:
**                                                              Background thread
** Parameters :
**                                                              void * - Arguments to thread
** Returns		:
**                                                              None
** Comments		:
**                                                              A background thread that keeps
tabs on the state of all the
**                                                              threads of the client.
**
****************************************************************
*/

void *bg_thread(void *argP)
{
		int
total_newo;
		int
total_tran_err;
		int
sleep_time                                      = 60000;

		char
logmsg[MAX_MSG_LEN];

		double                                                  time_diff1;
		double                                                  time_diff2;
		double                                                  tran_diff1;
		double                                                  tran_diff2;
		double                                                  prev_newo1;
		double                                                  prev_newo2;
		double                                                  err_diff1;
		double                                                  err_diff2;

		struct timespec			sleep_end;
		struct timespec			time_wait
= {60,0};

		struct timeval			cur_time;
		struct timeval			time_reported[2];

		struct timezone			tz;

		total_tran_count_t		tran_ct;
		total_tran_count_t		tran_reported[2];

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

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

		while(1)
		{
			check_threads(&tran_ct,NULL,NULL);

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

			gettimeofday(&cur_time, &tz);

			time_diff1	= time_diff_ms(&cur_time,&time_reported[0]);
			prev_newo1	= tran_reported[0].tran[NEW_ORDER_TRANS].num -
tran_reported[0].tran[NEW_ORDER_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[NEW_ORDER_TRANS].num -
tran_reported[1].tran[NEW_ORDER_TRANS].errs;
			tran_diff2	= total_newo - prev_newo2;
			err_diff2	= total_tran_err - tran_reported[1].errors;

			if(total_newo != 0 && tran_diff2 > 0)
			{
				sprintf(logmsg,"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.);
				err_printf(logmsg);

				// 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)
			{
				sprintf(logmsg,"bg_thread: errPM %.1f (last %.0f sec)\n",
                                                                        err_diff2 / time_diff2 * 60000,
time_diff2 / 1000.);
				err_printf(logmsg);
			}

			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);
		}

		return NULL;
}

/*
****************************************************************
** Name				: check_threads
** Description:
**                                                              Run though threads and check
status
** Parameters :
**                                                              total_tran_count_t * -
transactions
**                                                              int - number of threads?
**                                                              int * - threads initilized?
** Returns		:
**                                                              None
** Comments		:
**
****************************************************************
*/

void check_threads(total_tran_count_t *tran_ctP, int *num_threadsP, int *num_threadsInitP)
{
		char
		logmsg[MAX_MSG_LEN];
		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)
		{
			sprintf(logmsg,"bg_thread: %d errs: %d no, %d pa, %d os, %d sl\n",
                                                                        pClientInfo->errors,

pClientInfo->tran[NEW_ORDER_TRANS].errs,
```

```
                                                pClientInfo->tran[PAYMENT_TRANS].errs,
pClientInfo->tran[ORDER_STATUS_TRANS].errs,
                                                pClientInfo->tran[STOCK_TRANS].errs);
                        err_printf(logmsg);
                }

}
/*
****************************************************************
** Name                                 : start_bg_debug_thread
** Description:
**                                                              Start the background thread
** Parameters :
**                                                              None
** Returns              :
**                                                              None
** Comments             :
**
****************************************************************
*/

void start_bg_debug_thread()
{
                int                                             rc;

                char                                            logmsg[MAX_MSG_LEN];

                pthread_t               thread;
                pthread_attr_t attr;

                rc = pthread_attr_init(&attr);
                if(rc != OK)
                {
                        sprintf(logmsg,"start_bg_debug_thread: pthread_attr_init failed: %d\n", rc);
                        err_printf(logmsg);
                        return;
                }
                pthread_attr_setdetachstate(&attr,PTHREAD_CREATE_UNDETACHED);

                rc = pthread_create(&thread,&attr,bg_thread,(void *) NULL);
                if(rc != OK)
                {
                        sprintf(logmsg,"start_bg_debug_thread: pthread_create failed: %d\n", rc);
                        err_printf(logmsg);
                        return;
                }

                pthread_attr_destroy(&attr); rc = pthread_detach(thread);
                if(rc != OK)
                {
                        sprintf(logmsg,"start_bd_debug_thread: pthread_detach failed %d\n", rc);
                        err_printf(logmsg);
                        return;
                }

                return;
}

/*
****************************************************************
** Name                                 : client_status_report
** Description:
**                                                              Report client status
** Parameters :
**                                                              int - socket descriptor to RTE
** Returns              :
**                                                              None
** Comments             :
**
****************************************************************
*/

void *client_status_report(int fileno)
{
                                                int
                        cnt                                     = 0;

                                                char
                        buf[1024];
                                                char
                        cmd                                     = NULL;
                                                char
                        logmsg[MAX_MSG_LEN];

                static  struct timespec                         time_wait       = {60, 0};
                                                struct timeval
cur_time;
                                                struct timezone                                 tz;

                                                total_tran_count_t              tran_ct;
                                                tran_info_t
*curP;

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

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

                        // read next cmd from the socket
                        read(fileno,buf,1); cmd = buf[0];
                        if(cmd=='q') break;

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

                        // The tpcc_monitor has to read the data in the same order to get the correct
data.

                        sprintf(logmsg,"\n"); prefix_sprintf(buf,logmsg);
                        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[NEW_ORDER_TRANS].num-tran_ct.tran[NEW_ORDER_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));

                        curP = tran_ct.tran;
                        for(int i = 0;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,"....",4);
                }

                return NULL;
}

//////////////////////////////////////////////////////////////////////
```

tuxClientThread.hpp

```
/*
****************************************************************
** Project                      : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                                 : tuxClientThread.hpp
** Title                                : Tux client thread communicating with RTE
****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History      :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                             :
**
****************************************************************
*/

#ifndef __tuxClientThread_hpp__
#define __tuxClientThread_hpp__

#include <commonTuxDB2.hpp>
#include <tuxClientCommon.hpp>

//////////////////////////////////////////////////////////////////////
// Function Definitions
//////////////////////////////////////////////////////////////////////

char *thread_state_to_str(int state);

void set_client_debug_state(void *contextP,int state,int tran);
void print_rt_avg(total_tran_count_t *curP,total_tran_count_t *prevP,int type);
void check_threads(total_tran_count_t *tran_ctP, int *num_threadsP, int *num_threadsInitP);
void start_bg_debug_thread();
void *client_status_report(int fileno);

struct timeval *client_last_time(thread_descr_t *descrP);
```

```
#endif
////////////////////////////////////////////////////////////////////
```

Makefile

```
######################################################################
## Project                                  : DB/2 TPCC
## Component                : TUX Server
## Title                                                : tuxServer
## Copyright     : 2003 IBM Corporation
## Classification : All rights reserved
##
## History       :
##              Developed at IBM Austin by the AIX RS/6000
##              performance group.
##
## Comments      :
##
######################################################################

######################################################################
## Declare the module name & path & Local Vars
##                              NB the metafile name must not exceed eight characters and
##                                              is unprefiexed
######################################################################

PROGRAM                 = tuxServer
MODULE                  = /home/tpcc/toraus/db2kit/db2server
COMMON                  = /home/tpcc/toraus/db2kit/db2common
CLIENT                  = /home/tpcc/toraus/db2kit/db2client
EXEHOME                 = /home/tpcc/toraus/db2kit/db2inf
TUXINC                  = /usr/local/bea/tuxedo8.0/include
INSTALLDIR= $(EXEHOME)
MAKEFILE    = makefile
DB2LIBS                 = /usr/opt/db2_08_01/lib
DB2SQLKIT  = /home/tpcc/toraus/db2kit/db2inf/tpcccli.a

SETDEBUG    = -g
SETOPT      = -O3 -qinlglue -qproto -qunroll=8

######################################################################
## Compiler settings
######################################################################

CC                                              = xlC_r
CCC                                             = xlC_r
CFLAGS      = -I/home/tpcc/toraus/db2kit/db2server \
        -I /home/tpcc/toraus/db2kit/db2common \
        -I /home/tpcc/toraus/db2kit/db2client \
        -I /usr/local/bea/tuxedo8.0/include \
        -D__UNIX__ -D__AIXPOWER__ -D__DB2__ \
        -+ -qnotempinc -qcpluscmt -qchars=unsigned
CCFLAGS     = -I/home/tpcc/toraus/db2kit/db2server \
        -I /home/tpcc/toraus/db2kit/db2common \
        -I /home/tpcc/toraus/db2kit/db2client \
        -I /usr/local/bea/tuxedo8.0/include \
        -D__UNIX__ -D__AIXPOWER__ -D__DB2__ \
        -+ -qnotempinc -qcpluscmt -qchars=unsigned
LPATH       =
LIBS                                            = -lc -lm -lsvld -ls -libmuis
LFLAGS                  = -lc -lm -lsvld -ls -libmuis -bh:5

######################################################################
## define what componects are required for module
######################################################################

######################################################################
## Module Source files (.cpp)
##                              NB the source files must be listed in the same order as
##                                              the object files
######################################################################

PRGSRCS                 = \

$(CLIENT)/tuxClientNullDB.cpp \
                                                                        $(MODULE)/tuxServer.cpp

######################################################################
## Module Object files (.o)
##                              NB the object files must be listed in the same order as
##                                              the source files
######################################################################

PRGOBJS                 = \
                                                        tuxClientNullDB.o \
                                                        tuxServer.o

######################################################################
## standard makefile rules
######################################################################

######################################################################
## Setup lists of files for processing
######################################################################

SOURCECODE              = $(PRGSRCS)

OBJECTCODE              = $(PRGOBJS)

######################################################################
## File Sufixes
##                              .sxx  - sqc metafile
##                              .sqc            - imbeded sql
##                              .cpp            - c++ source code
##                              .c                              - c source code
##                              .o                              - object code
######################################################################

.SUFFIXES: .cpp .c .o

######################################################################
## Build Targets
##                              all                             - build module
##                              link                            - link module
##                              strip                           - strip module
##                              install         - install module
##                              clean                           - clean module working directory
##                              depends         - update makefile dependencies
######################################################################

all: $(PROGRAM)

$(PROGRAM): $(OBJECTCODE)
                ar -v -r $(PROGRAM).a $(PRGOBJS)
                buildserver -o $(PROGRAM) -f $(PROGRAM).a -f $(DB2LIBS)/libdb2.a -f
$(DB2SQLKIT) -s INFO:tux_db_info -s ORDER:tux_new_order -s PAYMENT:tux_payment -s
STATUS:tux_order_status -s DELIVERY:tux_delivery -s STOCK:tux_stock_level

link:
                ar -v -r $(PROGRAM).a $(PRGOBJS)
                buildserver -o $(PROGRAM) -f $(PROGRAM).a -f $(DB2SQLKIT) -f
$(DB2LIBS)/libdb2.a -s INFO:tux_db_info -s ORDER:tux_new_order -s PAYMENT:tux_payment -s
STATUS:tux_order_status -s DELIVERY:tux_delivery -s STOCK:tux_stock_level

strip:
                @-strip $(PROGRAM)
                @-strip $(PROGRAM).a

install:
                @-rm -f $(INSTALLDIR)/$(PROGRAM)
                @-rm -f $(INSTALLDIR)/$(PROGRAM).a
                @-cp -f $(PROGRAM) $(INSTALLDIR)
                @-cp -f $(PROGRAM).a $(INSTALLDIR)
                @-chmod 755 $(INSTALLDIR)/$(PROGRAM)
                @-chmod 755 $(INSTALLDIR)/$(PROGRAM).a

clean:
                @-rm -f $(OBJECTCODE)
                @-rm -f $(PROGRAM)
                @-rm -f $(PROGRAM).a

depends:
                @-/home/tpcc/toraus/db2kit/db2inf/depend -m$(MAKEFILE) $(INC) $(SOURCECODE)

######################################################################
##              Source to Object (.cpp -> .o)
######################################################################

.cpp.o:
                $(CCC) -q32 -c $< -o $@ $(INC) $(CCFLAGS)

######################################################################
##              Source to Object (.c -> .o)
######################################################################

.c.o:
                $(CCC) -q32 -c $< -o $@ $(INC) $(CCFLAGS)

######################################################################
##              Makefile End
######################################################################

# DO NOT DELETE THIS LINE -- dependency generator depends on it. Every after this line is replaced

tuxClientNullDB.o: \
                /home/tpcc/toraus/db2kit/db2client/tuxClientNullDB.cpp

tuxServer.o: \
                /home/tpcc/toraus/db2kit/db2server/tuxServer.cpp \
                ./tuxServer.hpp
```

tuxServer.cpp

```
/*
****************************************************************
** Project                      : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                                 : tuxServer.cpp
```

```
** Title                                    : Server for tux application
*****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History       :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group.
**
** Comments                                  :
**                                                                    RUN under
TUX
** Notes                                                  :
**
*****************************************************************
*/

#include <tuxServer.hpp>

//////////////////////////////////////////////////////////////////
// Globals
//////////////////////////////////////////////////////////////////

int                                         null_db
                =               0;

int                                         delivery_count                          =
0;
FILE                                        *delivery_log                           =
NULL;
FILE                                        *dump_log
=               NULL;

//////////////////////////////////////////////////////////////////
// Helpers
//////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name       : tux_server_response
** Description :
**              Set transaction header return codes
** Parameters  :
**              data_header - transaction data header
**              int - return code
**              int - sql_code
** Returns     :
**              None
** Comments    :
**
*****************************************************************
*/

void tux_server_response(data_header *header,int rc,int sql_code)
{
                //set return codes
                header->dtype       =  0;
                header->returncode   =  rc;
                header->sql_code     =  sql_code;
                header->isam_code    =  0;
                header->num_rms      =  0;
                header->stats        =  0;
                return;
}

//////////////////////////////////////////////////////////////////
// TUX interface calls
//////////////////////////////////////////////////////////////////

/*
*****************************************************************
** Name       : tpsvrinit
** Description :
**              TUX server initilize
** Parameters  :
**              None
** Returns     :
**                                                                    None
** Comments    :
**
** Notes       :
**                                                                    We can only hope that the
database connection
**                                                                    is sucessful as there is no return
code provided.
**
*****************************************************************
*/

extern "C" int tpsvrinit(int argc,char **argv)
{
                int             rc = 0;
                char            tpcc_database[MAX_DATABASE_NAME_LEN];
                char            file_name[MAX_STR_LEN];

                #if DUMP_DATA

sprintf(file_name,"%s/tuxServerDump%d.log",getenv("TPCC_DELIVERY"),getpid());
                                dump_log=fopen(file_name,"w");
                #endif

                // Set null database attribute
                char *null_database = getenv("TPCC_NULL");
                null_db = atoi(null_database);

                int del_srv                                     =               0;
                int bypass_conn                 =               0;
                extern char *optarg;
                while ((rc = getopt(argc,argv,"dx")) != EOF)
                {
                        switch (rc)
                        {
                                case 'd':
                                        del_srv=1;
                                        break;
                                case 'x':
                                        bypass_conn=1;
                                default:
                                        break;
                        }
                }

                if ( del_srv )
                {
                        // Set the delivery log path
                        if(getenv("TPCC_DELIVERY"))
                        {
                                //Set file name

sprintf(file_name,"%s/tuxServerDelivery%d.log",getenv("TPCC_DELIVERY"),getpid());

                                delivery_log=fopen(file_name,"w");
                                if(!delivery_log)
                                {
                                        printf("Fatal Error: Unable to open Delivery log
file %s\n",file_name);
                                        exit(1);
                                }
                        }
                        else
                        {
                                printf("Fatal Error: TPCC_DELIVERY environment variable
not set\n");
                                exit(1);
                        }
                }

                // we need to connect to the database
                if(!null_db && !bypass_conn)
                {
                        if(getenv("TPCC_DB"))
                                strcpy(tpcc_database,getenv("TPCC_DB"));
                        else strcpy(tpcc_database,"tpcc");

                        #if DUMP_DATA
                                fprintf(dump_log,"Calling connect_to_TM\n");
                                fflush(dump_log);
                        #endif

                        // make database connection.
                        rc=connect_to_TM(tpcc_database);

                        if (rc != OK)
                        {
                                printf("Fatal Error: Unable to establish connection to %s
database, return code %d\n",tpcc_database,rc);
                                #if DUMP_DATA
                        fprintf(dump_log,"Unable to connect to database, return code %d\n",rc);
                fflush(dump_log);
                #endif
                                return 0;
                        }
                }

                return 0;
}

/*
*****************************************************************
** Name       : tpsvrdone
** Description :
**              TUX server unload
** Parameters  :
**              None
** Returns     :
**                                                                    None
** Comments    :
**
** Notes       :
**
*****************************************************************
*/

extern "C" void tpsvrdone(void)
{
                int rc = 0;

                #if DUMP_DATA
```

```
                        fprintf(dump_log,"Calling disconnect_from_TM()\n");
                        fflush(dump_log);
                #endif

                // Disconnect from database
                if(!null_db)
                {
                        rc = disconnect_from_TM();
                        if (rc != OK)
                        {
                                printf("Error: Unable to disconnect from database, return code
%d\n",rc);

                                #if DUMP_DATA
                        fprintf(dump_log,"Unable to disconnect from database, return code %d\n",rc);
                        fflush(dump_log);
                        #endif
                        }
                }
                // Close delivery log file
                if(delivery_log != NULL)
                        fclose(delivery_log);

                // Close dump log file
                if(dump_log != NULL)
                        fclose(dump_log);

                return;
}

/*
*****************************************************************
** Name       : tux_db_info
** Description :
**             Get database info
** Parameters  :
**             TPSVCINFO * - request
** Returns     :
**                                                             None
** Comments    :
**
** Notes       :
**
*****************************************************************
*/

extern "C" void tux_db_info(TPSVCINFO *rqst)
{
                dbInfo_data_t *pData = (dbInfo_data_t *) rqst->data;

                if(!null_db)
                {
                        pData->first_wh                 = MIN_WAREHOUSE;
                        pData->last_wh                          = MAX_WAREHOUSE;

                        pData->server_id                = 0;
                }
                else tux_db_info_null_db(pData);

                tux_server_response(&pData->header,0,0);
                tpreturn(TPSUCCESS,0,rqst->data,sizeof(dbInfo_data_t),0);
                return;
}

/*
*****************************************************************
** Name       : tux_new_order
** Description :
**             New order
** Parameters  :
**             TPSVCINFO * - request
** Returns     :
**                                                             None
** Comments    :
**
** Notes       :
**
*****************************************************************
*/

extern "C" void tux_new_order(TPSVCINFO *rqst)
{
                newOrder_data_t *pData = (newOrder_data_t *) rqst->data;

                if(!null_db)
                        neword_sql(&pData->in,&pData->out);
                else tux_new_order_null_db(pData);

                #if DUMP_DATA
                        if (pData->out.s_transtatus != OK)
                        {
                                fprintf(dump_log,"New Order SQL Error: %d
\n",pData->out.s_transtatus);
                                fflush(dump_log);
                        }
                #endif

                tux_server_response(&pData->header,pData->out.s_transtatus,pData->out.s_transtatus);
                tpreturn(TPSUCCESS,0,rqst->data,sizeof(newOrder_data_t),0);
                return;
}

/*
*****************************************************************
** Name       : tux_payment
** Description :
**             Payment
** Parameters  :
**             TPSVCINFO * - request
** Returns     :
**                                                             None
** Comments    :
**
** Notes       :
**
*****************************************************************
*/

extern "C" void tux_payment(TPSVCINFO *rqst)
{
                payment_data_t *pData = (payment_data_t *) rqst->data;

                #if DUMP_DATA
                        if (pData->in.s_H_AMOUNT == 0)
                        {
                                fprintf(dump_log,"Invalid payment from client,
amount:%lld\n",pData->in.s_H_AMOUNT);
                                fflush(dump_log);
                        }
                #endif

                if(!null_db)
                {
                        payment_sql(&pData->in,&pData->out);
                }
                else tux_payment_null_db(pData);

                #if DUMP_DATA
                        if (pData->out.s_transtatus != OK)
                        {
                                fprintf(dump_log,"Payment SQL Error: %d
\n",pData->out.s_transtatus);
                                fflush(dump_log);
                        }
                #endif

                tux_server_response(&pData->header,pData->out.s_transtatus,pData->out.s_transtatus);
                tpreturn(TPSUCCESS,0,rqst->data,sizeof(payment_data_t),0);
                return;
}

/*
*****************************************************************
** Name       : tux_order_status
** Description :
**             Order status
** Parameters  :
**             TPSVCINFO * - request
** Returns     :
**                                                             None
** Comments    :
**
** Notes       :
**
*****************************************************************
*/

extern "C" void tux_order_status(TPSVCINFO *rqst)
{
                orderStatus_data_t *pData = (orderStatus_data_t *) rqst->data;

                if(!null_db)
                        ordstat_sql(&pData->in,&pData->out);
                else tux_order_status_null_db(pData);

                #if DUMP_DATA
                        if (pData->out.s_transtatus != OK)
                        {
                                fprintf(dump_log,"Order Status SQL Error: %d
\n",pData->out.s_transtatus);
                                fflush(dump_log);
                        }
                #endif

                tux_server_response(&pData->header,pData->out.s_transtatus,pData->out.s_transtatus);
                tpreturn(TPSUCCESS,0,rqst->data,sizeof(orderStatus_data_t),0);
                return;
}

/*
*****************************************************************
** Name       : tux_delivery
** Description :
**             Delivery
** Parameters  :
**             TPSVCINFO * - request
** Returns     :
**                                                             None
```

```
** Comments    :
**
** Notes       :
**
*****************************************************************
*/

extern "C" void tux_delivery(TPSVCINFO *rqst)
{
        delivery_data_t *pData = (delivery_data_t *) rqst->data;
        struct timeval  end_queue;
        struct timeval          end_process;
        struct timezone         tp;

        char
order_ids[MAX_STR_LEN];

        delivery_count++;

        if(!null_db)
        {
                gettimeofday(&end_queue,&tp);

                delivery_sql(&pData->in,&pData->out);

                gettimeofday(&end_process,&tp);
        }
        else
        {
                gettimeofday(&end_queue,&tp);

                tux_delivery_null_db(pData);

                gettimeofday(&end_process,&tp);
        }


        #if DUMP_DATA
                if (pData->out.s_transtatus != OK)
                {
                        fprintf(dump_log,"Delivery SQL Error: %d
\n",pData->out.s_transtatus);
                        fflush(dump_log);
                }
        #endif

        tux_server_response(&pData->header,pData->out.s_transtatus,pData->out.s_transtatus);

        if (pData->out.s_transtatus == OK )
        {
                //list out the order ids
                int bytes_written                       =       0;
                char *str_ptr                           =       order_ids;

                for(int district_id=0; district_id < DISTRICTS_PER_WAREHOUSE;
district_id++)
                {
                        if ( pData->out.s_O_ID[district_id] == 0 )
                        {
                                bytes_written=sprintf(str_ptr,"\nD_ID %d has
no new orders",district_id);
                                str_ptr+=bytes_written;
                        }
                        else
                        {
                                bytes_written=sprintf(str_ptr,"%d
",pData->out.s_O_ID[district_id]);
                                str_ptr+=bytes_written;
                        }
                }
        }
        else
        {
                sprintf(order_ids,"\nDelivery transaction failed");
        }
        //log the transaction
        fprintf(delivery_log,DELIVERY_LOG_SUCCESS_STR,
                                                                delivery_count,
                                                        // delivery transaction number
pData->header.start_time.sec,           // tux client enqueue time
pData->header.start_time.usec/1000,     //
                                                                end_queue.tv_sec,
                                                        // tux server dequeue time
end_queue.tv_usec/1000,                                         //
pData->in.s_W_ID,                                                       //
warehouse id
pData->in.s_O_CARRIER_ID,                               // carrier id
                                                                order_ids,
                                                                // string of carrier
ids
end_process.tv_sec,                                             // final transaction
end time
end_process.tv_usec/1000);
                fflush(delivery_log);

                tpreturn(TPSUCCESS,0,rqst->data,sizeof(delivery_data_t),0);

                return;
}

/*
*****************************************************************
** Name       : tux_stock_level
** Description :
**              Stock level
** Parameters  :
**              TPSVCINFO * - request
** Returns     :
**                                                                      None
** Comments    :
**
** Notes       :
**
*****************************************************************
*/

extern "C" void tux_stock_level(TPSVCINFO *rqst)
{
        stockLevel_data_t *pData = (stockLevel_data_t *) rqst->data;

        if(!null_db)
        {
                stocklev_sql(&pData->in,&pData->out);
        }
        else tux_stock_level_null_db(pData);

        #if DUMP_DATA
                if (pData->out.s_transtatus != OK)
                {
                        fprintf(dump_log,"Stock SQL Error: %d
\n",pData->out.s_transtatus);
                        fflush(dump_log);
                }
        #endif

        tux_server_response(&pData->header,pData->out.s_transtatus,pData->out.s_transtatus);
        tpreturn(TPSUCCESS,0,rqst->data,sizeof(stockLevel_data_t),0);
        return;
}

/*
*****************************************************************
** Name       : hex_dump_stock
** Description :
**              dump the in and out structures in hex
** Parameters  :
**              stockLevel_data_t * - main structure
** Returns     :
**              None
** Comments    :
**              None
**
*****************************************************************
*/
void hex_dump_stock(stockLevel_data_t *pdata)
{
        //print hex structure out in order: main address,in address, contents of in struct, header
address
  //header contents, out address, contents of out address

  fprintf(dump_log,"START DUMP\n");

  //main address
  fprintf(dump_log,"/nstock struct:%x\n",&pdata);

        //in struct
  fprintf(dump_log,"struct in_stocklev_struct:%x",&pdata->in);
        fprintf(dump_log,"len:%x      value:%x\n",&pdata->in.len,pdata->in.len);
        fprintf(dump_log,"pad:%x      value:%x\n",&pdata->in.pad,pdata->in.pad);
        fprintf(dump_log,"s_W_ID:%x    value:%x\n",&pdata->in.s_W_ID,pdata->in.s_W_ID);
  fprintf(dump_log,"s_D_ID:%x    value:%x\n",&pdata->in.s_D_ID,pdata->in.s_D_ID);
  fprintf(dump_log,"s_threshold:%x    value:%x\n",&pdata->in.s_threshold,pdata->in.s_threshold);

        //header struct
  fprintf(dump_log,"dtype:%x\n",&pdata->header.dtype);
  fprintf(dump_log,"returncode:%x    value:%x\n",&pdata->header.returncode,pdata->header.returncode);
  fprintf(dump_log,"sql_code:%x    value:%x\n",&pdata->header.sql_code,pdata->header.sql_code);
  fprintf(dump_log,"isam_code:%x    value:%x\n",&pdata->header.isam_code,pdata->header.isam_code);
  fprintf(dump_log,"num_rms:%x    value:%x\n",&pdata->header.num_rms,pdata->header.num_rms);
  fprintf(dump_log,"stats:%x    value:%x\n",&pdata->header.stats,pdata->header.stats);
  fprintf(dump_log,"start_time:%x\n",&pdata->header.start_time);
  fprintf(dump_log,"sec:%x    value:%x\n",&pdata->header.start_time.sec,pdata->header.start_time.sec);
  fprintf(dump_log,"usec:%x
value:%x\n",&pdata->header.start_time.usec,pdata->header.start_time.usec);
  fprintf(dump_log,"end_time:%x\n",&pdata->header.end_time);
  fprintf(dump_log,"sec:%x    value:%x\n",&pdata->header.end_time.sec,pdata->header.end_time.sec);
  fprintf(dump_log,"usec:%x    value:%x\n",&pdata->header.end_time.usec,pdata->header.end_time.usec);

        //out struct
  fprintf(dump_log,"struct out_stocklev_struct:%x\n",&pdata->out);
        fprintf(dump_log,"len:%x      value:%x\n",&pdata->out.len,pdata->out.len);
```

```
                fprintf(dump_log,"pad:%x     value:%x\n",&pdata->out.pad,pdata->out.pad);
  fprintf(dump_log,"s_transtatus:%x    value:%x\n",&pdata->out.s_transtatus,pdata->out.s_transtatus);
  fprintf(dump_log,"deadlocks:%x    value:%x\n",&pdata->out.deadlocks,pdata->out.deadlocks);
  fprintf(dump_log,"s_low_stock:%x    value:%x\n",&pdata->out.s_low_stock,pdata->out.s_low_stock);

  fprintf(dump_log,"END DUMP\n");

}

/*
****************************************************************
** Name       : hex_dump_payment
** Description :
**             dump the in and out structures in hex
** Parameters  :
**             stockLevel_data_t * - main structure
** Returns     :
**             None
** Comments    :
**             None
**
****************************************************************
*/
void hex_dump_payment(payment_data_t *pdata)
{
  //print hex structure out in order: main address,in address, contents of in struct, header address
  //header contents, out address, contents of out address

  fprintf(dump_log,"START DUMP\n");

  //main address
  fprintf(dump_log,"\npayment struct:%x\n",&pdata);

  //in struct
  fprintf(dump_log,"struct in_stocklev_struct:%x",&pdata->in);
  fprintf(dump_log,"len:%x  value:%x\n",&pdata->in.len,pdata->in.len);
  fprintf(dump_log,"pad:%x  value:%x\n",&pdata->in.pad,pdata->in.pad);
  fprintf(dump_log,"s_H_DATE_time:%x
value:%x\n",&pdata->in.s_H_DATE_time,pdata->in.s_H_DATE_time);
  fprintf(dump_log,"s_H_AMOUNT:%x
value:%lld\n",&pdata->in.s_H_AMOUNT,pdata->in.s_H_AMOUNT);
  fprintf(dump_log,"s_W_ID:%x    value:%x\n",&pdata->in.s_W_ID,pdata->in.s_W_ID);
  fprintf(dump_log,"s_C_ID:%x    value:%x\n",&pdata->in.s_C_ID,pdata->in.s_C_ID);
  fprintf(dump_log,"s_C_W_ID:%x  value:%x\n",&pdata->in.s_C_W_ID,pdata->in.s_C_W_ID);
                fprintf(dump_log,"s_C_ID:%x
value:%x\n",&pdata->in.s_C_ID,&pdata->in.s_C_ID);
                fprintf(dump_log,"s_C_D_ID:%x
value:%x\n",&pdata->in.s_C_D_ID,pdata->in.s_C_D_ID);
  fprintf(dump_log,"s_D_ID:%x    value:%x\n",&pdata->in.s_D_ID,pdata->in.s_D_ID);

  //header struct
  fprintf(dump_log,"dtype:%x\n",&pdata->header.dtype);
  fprintf(dump_log,"returncode:%x    value:%x\n",&pdata->header.returncode,pdata->header.returncode);
  fprintf(dump_log,"sql_code:%x    value:%x\n",&pdata->header.sql_code,pdata->header.sql_code);
  fprintf(dump_log,"isam_code:%x    value:%x\n",&pdata->header.isam_code,pdata->header.isam_code);
  fprintf(dump_log,"num_rms:%x    value:%x\n",&pdata->header.num_rms,pdata->header.num_rms);
  fprintf(dump_log,"stats:%x    value:%x\n",&pdata->header.stats,pdata->header.stats);
  fprintf(dump_log,"start_time:%x\n",&pdata->header.start_time);
  fprintf(dump_log,"sec:%x    value:%x\n",&pdata->header.start_time.sec,pdata->header.start_time.sec);
  fprintf(dump_log,"usec:%x
value:%x\n",&pdata->header.start_time.usec,pdata->header.start_time.usec);
  fprintf(dump_log,"end_time:%x\n",&pdata->header.end_time);
  fprintf(dump_log,"sec:%x    value:%x\n",&pdata->header.end_time.sec,pdata->header.end_time.sec);
  fprintf(dump_log,"usec:%x    value:%x\n",&pdata->header.end_time.usec,pdata->header.end_time.usec);

  //out struct
  fprintf(dump_log,"struct out_stocklev_struct:%x\n",&pdata->out);
  fprintf(dump_log,"len:%x  value:%x\n",&pdata->out.len,pdata->out.len);
  fprintf(dump_log,"pad:%x  value:%x\n",&pdata->out.pad,pdata->out.pad);
  fprintf(dump_log,"s_H_DATE_time:%x
value:%x\n",&pdata->out.s_H_DATE_time,pdata->out.s_H_DATE_time);
  fprintf(dump_log,"s_C_SINCE_time:%x
value:%x\n",&pdata->out.s_C_SINCE_time,pdata->out.s_C_SINCE_time);
  fprintf(dump_log,"s_C_CREDIT_LIM:%x
value:%x\n",&pdata->out.s_C_CREDIT_LIM,pdata->out.s_C_CREDIT_LIM);
  fprintf(dump_log,"s_C_BALANCE:%x
value:%x\n",&pdata->out.s_C_BALANCE,pdata->out.s_C_BALANCE);
  fprintf(dump_log,"s_C_DISCOUNT:%x
value:%x\n",&pdata->out.s_C_DISCOUNT,pdata->out.s_C_DISCOUNT);
  fprintf(dump_log,"s_C_ID:%x    value:%x\n",&pdata->out.s_C_ID,pdata->out.s_C_ID);
  fprintf(dump_log,"s_transtatus:%x    value:%x\n",&pdata->out.s_transtatus,pdata->out.s_transtatus);
  fprintf(dump_log,"deadlocks:%x    value:%x\n",&pdata->out.deadlocks,pdata->out.deadlocks);

  fprintf(dump_log,"END DUMP\n");
                fflush(dump_log);

}

/*
****************************************************************
** Name                                : hex_dump_delivery
**             Description :
**                                                                dump the in and out structures
in hex
**             Parameters     :
**                                                                delivery_data_t * - main
structure
**             Returns                :
**                                                  None
** Comments                :
**                                                                                  None
**
****************************************************************
*/
void hex_dump_delivery(delivery_data_t *pdata)
{
                //print hex structure out in order: main address,in address, contents of in struct, header
address
                //header contents, out address, contents of out address

                fprintf(dump_log,"START DUMP\n");

                //main address
                fprintf(dump_log,"\ndelivery struct:%x\n",&pdata);

                //in struct
                fprintf(dump_log,"struct in_delivery_struct:%x\n",&pdata->in);
                fprintf(dump_log,"len:%x      value:%x\n",&pdata->in.len,pdata->in.len);
                fprintf(dump_log,"pad:%x      value:%x\n",&pdata->in.pad,pdata->in.pad);
                fprintf(dump_log,"s_W_ID:%x  value:%x\n",&pdata->in.s_W_ID,pdata->in.s_W_ID);
                fprintf(dump_log,"s_O_CARRIER_ID:%x
value:%x\n",&pdata->in.s_O_CARRIER_ID,pdata->in.s_O_CARRIER_ID);
                fprintf(dump_log,"s_O_DELIVERY_D_time:%x
value:%x\n",&pdata->in.s_O_DELIVERY_D_time,pdata->in.s_O_DELIVERY_D_time);

                //header struct
                fprintf(dump_log,"dtype:%x  value:%x\n",&pdata->header.dtype,pdata->header.dtype);
                fprintf(dump_log,"returncode:%x
value:%x\n",&pdata->header.returncode,pdata->header.returncode);
                fprintf(dump_log,"sql_code:%x
value:%x\n",&pdata->header.sql_code,pdata->header.sql_code);
                fprintf(dump_log,"isam_code:%x
value:%x\n",&pdata->header.isam_code,pdata->header.isam_code);
                fprintf(dump_log,"num_rms:%x
value:%x\n",&pdata->header.num_rms,pdata->header.num_rms);
                fprintf(dump_log,"stats:%x  value:%x\n",&pdata->header.stats,pdata->header.stats);
  fprintf(dump_log,"start_time:%x\n",&pdata->header.start_time);
                fprintf(dump_log,"sec:%x
value:%x\n",&pdata->header.start_time.sec,pdata->header.start_time.sec);
                fprintf(dump_log,"usec:%x
value:%x\n",&pdata->header.start_time.usec,pdata->header.start_time.usec);
                fprintf(dump_log,"end_time:%x  \n",&pdata->header.end_time);
                fprintf(dump_log,"sec:%x
value:%x\n",&pdata->header.end_time.sec,pdata->header.end_time.sec);
                fprintf(dump_log,"usec:%x
value:%x\n",&pdata->header.end_time.usec,pdata->header.end_time.usec);

                //out struct
                fprintf(dump_log,"struct out_delivery_struct:%x\n",&pdata->out);
                fprintf(dump_log,"len:%x      value:%x\n",&pdata->out.len,pdata->out.len);
                fprintf(dump_log,"pad:%x      value:%x\n",&pdata->out.pad,pdata->out.pad);
                fprintf(dump_log,"s_transtatus:%x
value:%x\n",&pdata->out.s_transtatus,pdata->out.s_transtatus);
                fprintf(dump_log,"deadlocks:%x
value:%x\n",&pdata->out.deadlocks,pdata->out.deadlocks);
                fprintf(dump_log,"s_O_ID[10]:%x  value:%x\n",&pdata->out.s_O_ID,pdata->out.s_O_ID);

                fprintf(dump_log,"END DUMP\n");
}

//////////////////////////////////////////////////////////////////
```

tuxServer.hpp

```
/*
****************************************************************
** Project                      : AIX DB/2 TPC-C
** Component : TPC-C/Client
** Name                                 : tuxServer.hpp
** Title                                : Tux Client server
****************************************************************
** Copyright (c) IBM US - AUSTIN 2003
** Classification : IBM Internal Use Only
**
** History       :
**             Developed at IBM Austin by the AIX RS/6000
**             performance group.
**
** Comments                             :
**
****************************************************************
*/

#ifndef __tuxServer_hpp__
#define __tuxServer_hpp__

#include <commonTuxDB2.hpp>
#include <tuxClientNullDB.hpp>

//////////////////////////////////////////////////////////////////
// externals
//////////////////////////////////////////////////////////////////
```

```
extern int                                      null_db;

////////////////////////////////////////////////////////////////
// Definitions
////////////////////////////////////////////////////////////////

#define MAX_DATABASE_NAME_LEN                   256

#define DELIVERY_LOG_SUCCESS_STR    "--Tran %d Queue %d.%03d Start %d.%03d\nW_ID: %d
CARRIER_ID: %d %s\nend-time: %d.%03d\n"
#define DELIVERY_LOG_SKIPPED_STR    "D_ID %d has no new orders"


////////////////////////////////////////////////////////////////
// Debug
////////////////////////////////////////////////////////////////
#define DUMP_DATA                               0

////////////////////////////////////////////////////////////////
// Function Definitions
////////////////////////////////////////////////////////////////

extern "C" void tux_db_info(TPSVCINFO *rqst);
extern "C" void tux_new_order(TPSVCINFO *rqst);
extern "C" void tux_payment(TPSVCINFO *rqst);
extern "C" void tux_order_status(TPSVCINFO *rqst);
extern "C" void tux_delivery(TPSVCINFO *rqst);
extern "C" void tux_stock_level(TPSVCINFO *rqst);

extern "C" void tux_open_delivery_log(char *filename);
extern "C" void tux_close_delivery_log(FILE *fileptr);

void hex_dump_delivery(delivery_data_t *pdata);
void hex_dump_stock(stockLevel_data_t *pdata);
void hex_dump_payment(payment_data_t *pdata);

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


#endif
```

commonTuxDB2.hpp

```
/*
*********************************************************************
** Project                  : AIX DB/2 TPC-C
** Component: TPC-C/Client
** Name                     : comonTuxDB2.hpp
** Title                    : Common header file
*********************************************************************
** Copyright (c) IBM US - AUSTIN 2000
** Classification           : IBM Internal Use Only
**
** History                  :
**                                                          09/03/2002
ADunshea Initial version.
**
** Comments                 :
**
*********************************************************************
*/

#ifndef __commonTuxDB2_hpp__
#define __commonTuxDB2_hpp__

////////////////////////////////////////////////////////////////
// Include files
////////////////////////////////////////////////////////////////

#include <pthread.h>
#include <stdio.h>
#include <string.h>
#include <strings.h>
#include <stdarg.h>
#include <sys/stat.h>
#include <errno.h>
#include <math.h>
#include <time.h>
#include <fcntl.h>
#include <termio.h>
#include <unistd.h>
#include <sys/ipc.h>
#include <termios.h>
#include <sys/socket.h>
#include <sys/socketvar.h>
#include <netinet/in.h>
#include <netinet/tcp.h>
#include <netdb.h>
#include <arpa/inet.h>

#include <atmi.h>

////////////////////////////////////////////////////////////////
// DB2 kit interface
////////////////////////////////////////////////////////////////
typedef int16    sqlint16;
typedef int32    sqlint32;
typedef int64    sqlint64;

#include <db2tpcc.h>

////////////////////////////////////////////////////////////////
// Common definitions
////////////////////////////////////////////////////////////////

#define OK                                      0
#define ERR                                     -1
#define CANCEL                                  -1
#define SUCCESS_CODE    0
#define RPC_ERROR                       -2

#define CHAR_NULL                       '\0'

////////////////////////////////////////////////////////////////
// fixed size data types
// Define platform inderpendent data types
////////////////////////////////////////////////////////////////

typedef int8                                    INT8b;
typedef int16                                   INT16b;
typedef int32                                   INT32b;
typedef int64                                   INT64b;

typedef u_int8                                  UINT8b;
typedef u_int16                                 UINT16b;
typedef u_int32                                 UINT32b;
typedef u_int64                                 UINT64b;

typedef char                                    BYTE8b;

typedef double                                  DOUBLE;

////////////////////////////////////////////////////////////////
// Transaction Definitions
////////////////////////////////////////////////////////////////

#define NAME_LENGTH                                     32

#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 MAX_OL
15
#define MAX_ITEMS                                               15

#define YEAR_TO_DATE                                    1
#define YEAR_TO_SECOND                                  2
#define COLLECTOR                                               1
#define D_PER_W                                                 10

#define MAX_STR_LEN                                     255
#define DATETIME_LEN                                    19

#define ERR_BAD_ITEM_ID                                 1
#define INVALID_ITEM                                    100

#define NEW_ORDER_INTERFACE             1
#define PAYMENT_INTERFACE                       2
#define ORDER_STATUS_INTERFACE  4
#define DELIVERY_INTERFACE                      8
#define STOCK_INTERFACE                                 16
#define ONLINE_INTERFACES                       23
#define ALL_INTERFACE                           65535

#define NEW_ORDER_TRANS                                 1
#define PAYMENT_TRANS                           2
#define ORDER_STATUS_TRANS                      3
#define DELIVERY_TRANS                          4
#define STOCK_TRANS                                     5
#define MAX_TRAN_TYPE                           5

#define DISTRICTS_PER_WAREHOUSE  10

////////////////////////////////////////////////////////////////
// Log File Defines
////////////////////////////////////////////////////////////////
#define COMM_LOG        0
#define DEBUG_LOG       1
```

```
#define ERROR_LOG          0

#define CHECK_RC                        0
////////////////////////////////////////////////////////////////
// Client Interface header structures
////////////////////////////////////////////////////////////////

typedef struct
{
        INT32b                                          sec;
        INT32b                                          usec;

} header_time_t;

typedef struct
{
        INT16b                                          dtype;
        INT16b                                          returncode;
        INT32b                                          sql_code;
        INT32b                                          isam_code;
        INT32b                                          num_rms;
        INT16b                                          stats;

        header_time_t           start_time;
        header_time_t           end_time;
} data_header;

////////////////////////////////////////////////////////////////
// payment
////////////////////////////////////////////////////////////////

typedef struct
{
        INT32b                                                  inputName;

        in_payment_struct                       in;
        data_header                                     header;
        out_payment_struct      out;
} payment_data_t;

////////////////////////////////////////////////////////////////
// newOrder
////////////////////////////////////////////////////////////////

typedef struct
{
        in_neword_struct                        in;
        data_header                                     header;
        out_neword_struct                       out;
} newOrder_data_t;

////////////////////////////////////////////////////////////////
// orderStatus
////////////////////////////////////////////////////////////////

typedef struct
{
        INT32b                                                  inputName;

        in_ordstat_struct                       in;
        data_header                                     header;
        out_ordstat_struct      out;
} orderStatus_data_t;

////////////////////////////////////////////////////////////////
// stockLevel
////////////////////////////////////////////////////////////////

typedef struct
{
        in_stocklev_struct      in;
        data_header                                     header;
        out_stocklev_struct     out;
} stockLevel_data_t;

////////////////////////////////////////////////////////////////
// delivery
////////////////////////////////////////////////////////////////

typedef struct
{
        in_delivery_struct      in;
        data_header                                     header;
        out_delivery_struct     out;
} delivery_data_t;

////////////////////////////////////////////////////////////////
// dbInfo
////////////////////////////////////////////////////////////////

typedef struct
{
        data_header                                     header;
        INT32b                                          first_wh;
        INT32b                                          last_wh;
        INT32b                                          server_id;
} dbInfo_data_t;

////////////////////////////////////////////////////////////////
// Main structures
////////////////////////////////////////////////////////////////

typedef struct
{
        INT32b                                                  tran_type;
        union
        {
                newOrder_data_t                         new_order;
                payment_data_t                                  payment;
                orderStatus_data_t      order_status;
                delivery_data_t                         delivery;
                stockLevel_data_t                       stock_level;
        } data;
} tpcc_data_t;

////////////////////////////////////////////////////////////////
// Client Types
////////////////////////////////////////////////////////////////

typedef int                                                             trpc_status_t;

#endif
////////////////////////////////////////////////////////////////
```

db2tpcc.h

```
/*********************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*********************************************************************/

/*
 * db2tpcc.h - Macros and Miscellany
 */

#ifndef __DB2TPCC_H
#define __DB2TPCC_H

#include <sys/types.h>

#include "lval.h"

/* ********************************************************************** */
/* Transaction Return Codes (s_transtatus)                                */
/* ********************************************************************** */

#define INVALID_ITEM          100
#define TRAN_OK               0
#define FATAL_SQLERROR        -1

/* ********************************************************************** */
/* Definition of Unused and Bad Items                                     */
/* ********************************************************************** */
/* Define unused item ID to be 0. This allows the SUT to determine the    */
/* number of items in the order as required by 2.4.1.3 and 2.4.2.2 since  */
/* the assumption that any item with OL_I_ID = 0 is unused will be true.  */
/* This in turn requires that the value used for an invalid item is       */
/* equal to ITEMS + 1.                                                    */
/* ********************************************************************** */

#define INVALID_ITEM_ID (2 * ITEMS) + 1
#define UNUSED_ITEM_ID 0

#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES

/*********************************************************************/
/* NURand Constants                                                  */
/* C_C_LAST_RUN and C_C_LAST_LOAD must adhere to clause 2.1.6.       */
/*********************************************************************/
#define C_C_LAST_RUN    88
#define C_C_LAST_LOAD   173
#define C_C_ID                          127
#define C_OL_I_ID               1723
#define A_C_LAST                255
#define A_C_ID                          1023
#define A_OL_I_ID               8191

/*********************************************************************/
/* Transaction Type Identifiers                                      */
/*********************************************************************/

#define NEWORD_SQL   1
#define PAYMENT_SQL  2
```

```
#define ORDSTAT_SQL  3
#define DELIVERY_SQL 4
#define STOCKLEV_SQL 5

#define SPGENERAL_PAD 3
#define SPGENERAL_ADJUST sizeof(int16_t)

struct in_neword_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  struct   in_items_struct {
    int32_t  s_OL_I_ID;
    int32_t  s_OL_SUPPLY_W_ID;
    int16_t  s_OL_QUANTITY;
    int16_t  pad1[3];
  } in_item[15];
  int64_t  s_O_ENTRY_D_time;                /* init by SUT */
  int32_t  s_C_ID;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
  int16_t  s_O_OL_CNT;                      /* init by SUT */
  int16_t  s_all_local;
  int16_t  duplicate_items;
};

struct out_neword_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  struct   items_struct {
    int32_t  s_I_PRICE;
    int32_t  s_OL_AMOUNT;
    int16_t  s_S_QUANTITY;
    int16_t  pad2;
    char     s_I_NAME[25];
    char     s_brand_generic;
  } item[15];
  int64_t  s_O_ENTRY_D_time;
  int32_t  s_W_TAX;
  int32_t  s_D_TAX;
  int32_t  s_C_DISCOUNT;
  int32_t  s_total_amount;
  int32_t  s_O_ID;
  int16_t  s_O_OL_CNT;
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_C_LAST[17];
  char     s_C_CREDIT[3];
};

struct in_payment_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int64_t  s_H_DATE_time;   /* init by SUT */
  int64_t  s_H_AMOUNT;
  int32_t  s_W_ID;
  int32_t  s_C_W_ID;
  int32_t  s_C_ID;
  int16_t  s_C_D_ID;
  int16_t  s_D_ID;
  char     s_C_LAST[17];
};

struct out_payment_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int64_t  s_H_DATE_time;
  int64_t  s_C_SINCE_time;
  int64_t  s_C_CREDIT_LIM;
  int64_t  s_C_BALANCE;
  int32_t  s_C_DISCOUNT;
  int32_t  s_C_ID;
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_W_STREET_1[21];
  char     s_W_STREET_2[21];
  char     s_W_CITY[21];
  char     s_W_STATE[3];
  char     s_W_ZIP[10];
  char     s_D_STREET_1[21];
  char     s_D_STREET_2[21];
  char     s_D_CITY[21];
  char     s_D_STATE[3];
  char     s_D_ZIP[10];
  char     s_C_FIRST[17];
  char     s_C_MIDDLE[3];
  char     s_C_LAST[17];
  char     s_C_STREET_1[21];
  char     s_C_STREET_2[21];
  char     s_C_CITY[21];
  char     s_C_STATE[3];
  char     s_C_ZIP[10];
  char     s_C_PHONE[17];
  char     s_C_CREDIT[3];
  char     s_C_DATA[201];
};

struct in_ordstat_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_C_ID;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
  int16_t  pad1[3];
  char     s_C_LAST[17];
};

struct out_ordstat_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int64_t  s_C_BALANCE;
  int64_t  s_O_ENTRY_D_time;
  int32_t  s_C_ID;
  int32_t  s_O_ID;
  int16_t  s_O_CARRIER_ID;
  int16_t  s_ol_cnt;
  int16_t  pad1[2];
  struct   oitems_struct {
    int64_t  s_OL_DELIVERY_D_time;
    int32_t  s_OL_AMOUNT;
    int32_t  s_OL_I_ID;
    int32_t  s_OL_SUPPLY_W_ID;
    int16_t  s_OL_QUANTITY;
    int16_t  pad2;
  } item[15];
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_C_FIRST[17];
  char     s_C_MIDDLE[3];
  char     s_C_LAST[17];
};

struct in_delivery_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int64_t  s_O_DELIVERY_D_time;          /* init by SUT */
  int32_t  s_W_ID;
  int16_t  s_O_CARRIER_ID;
};

struct out_delivery_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_O_ID[10];
  int16_t  s_transtatus;
  int16_t  deadlocks;
};

struct in_stocklev_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_threshold;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
};

struct out_stocklev_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_low_stock;
  int16_t  s_transtatus;
  int16_t  deadlocks;
};

/* ********************************************************************** */
/* Transaction Prototypes                                                 */
/* ********************************************************************** */

#ifdef __cplusplus
extern "C" {
#endif

extern int neword_sql(struct in_neword_struct*, struct out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct out_stocklev_struct*);

#ifdef __cplusplus
}
#endif

/* ********************************************************************** */
/* Database Connect/Disconnect Prototypes                                 */
/* ********************************************************************** */

#ifdef __cplusplus
extern "C" {
#endif

extern int connect_to_TM(char*);
extern int disconnect_from_TM(void);

#ifdef __cplusplus
}
#endif


#endif // __DB2TPCC_H
```

lval.h

```
/* lval.h - generated automatically at 20030504.1326 */

#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 53200
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900

#endif // __LVAL_H
```

tpccdef

```
#ident          "@(#) /home/tpcc/db2kit/db2inf/tpccdef      $Revision: 1.3 $"

*RESOURCES
IPCKEY                  32769

DOMAINID   tpckit
MASTER                  tpcc
MAXACCESSERS            1000
MAXSERVERS              200
MAXSERVICES             200
MAXBUFSTYPE             200
MODEL                   SHM
LDBAL                   Y
OPTIONS                 NO_AA,NO_XA

*MACHINES
DEFAULT:
APPDIR="/home/tpcc/db2kit/db2inf"
TUXCONFIG="/home/tpcc/db2kit/db2inf/tpcccfg"
TUXDIR="/usr/local/bea/tuxedo8.0"

<HOSTNAME>              LMID=tpcc

*GROUPS
GROUP1
        LMID=tpcc    GRPNO=1     OPENINFO=NONE
GROUP2
        LMID=tpcc    GRPNO=2     OPENINFO=NONE
GROUP3
        LMID=tpcc    GRPNO=3     OPENINFO=NONE
GROUP4
        LMID=tpcc    GRPNO=4     OPENINFO=NONE
GROUP5
        LMID=tpcc    GRPNO=5     OPENINFO=NONE
GROUP6
        LMID=tpcc    GRPNO=6     OPENINFO=NONE

*SERVERS
DEFAULT:
                        CLOPT="-A" REPLYQ=Y

tuxServer       SRVGRP=GROUP1 SRVID=1001 MIN=26 MAX=28 CLOPT="-s
ORDER:tux_new_order" RQADDR=newo
tuxServer       SRVGRP=GROUP2 SRVID=2001 MIN=22 MAX=24 CLOPT="-s
PAYMENT:tux_payment" RQADDR=pmt
tuxServer       SRVGRP=GROUP3 SRVID=3001 MIN=3  MAX=8 CLOPT="-s
STATUS:tux_order_status" RQADDR=ostat
tuxServer       SRVGRP=GROUP4 SRVID=4001 MIN=12 MAX=13 CLOPT="-s
DELIVERY:tux_delivery -- -d" RQADDR=dlvry
tuxServer       SRVGRP=GROUP5 SRVID=5001 MIN=3  MAX=8 CLOPT="-s STOCK:tux_stock_level"
RQADDR=stklvl
tuxServer       SRVGRP=GROUP6 SRVID=6001 MIN=1  MAX=2  CLOPT="-s INFO:tux_db_info -- -x"

*SERVICES
INFO
ORDER
PAYMENT
STATUS
DELIVERY

STOCK
```

killit

```
#!/bin/ksh

if [ "$1" = "" ]
then
        exit
fi

ps -e | grep $1 | cut -c1-7 | while read name
do
        if [ "$name" != "" ]
        then
                kill -9 $name
                echo $name killed
        fi

done
```

A.2 Client Transaction Code

Makefile

```
#################################################################################
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#################################################################################

#
# Makefile - Makefile for Src.Srv
#

include $(TPCC_ROOT)/Makefile.config

# ###############################################################
# Preprocessor, Compiler and Linker Flags
# ###############################################################

BND_OPTS =                  GRANT PUBLIC \
                            MESSAGES $*.bnd.msg
PRP_OPTS =  BINDFILE \
                            EXPLAIN ALL \
                            MESSAGES $*.prep.msg \
                            $(DB2V8CD)

INCLUDE =       -I$(TPCC_SQLLIB)/include -I$(TPCC_ROOT)/include

CFLAGS =    $(CFLAGS_OS) $(INCLUDE) $(CFLAGS_DEBUG) \
                            -D$(DB2EDITION) -D$(DB2VERSION) \
                            -DSQLA_NOLINES -DLINT_ARGS

LDFLAGS =   $(LDFLAGS_STORP) $(LDFLAGS_LIB)

# ###############################################################
# File Collections
# ###############################################################

STORED_PROCS =  new ord del

UTIL_OBJ =  $(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT) \
                            $(TPCC_ROOT)/Src.Common/tpccdbg$(OBJEXT)

EXE =                       news ords dels

# ###############################################################
# User Targets
# ###############################################################

all:            connect explain catalog $(EXE) install plan disconnect

clean:          connect uncatalog unexplain disconnect
                - $(ERASE) $(TPCC_SPDIR)$(SLASH)news
                - $(ERASE) $(TPCC_SPDIR)$(SLASH)ords
                - $(ERASE) $(TPCC_SPDIR)$(SLASH)dels
                - $(ERASE) *.bnd *.msg *$(OBJEXT) $(EXE) tpcc_all_sql.c
                - $(ERASE) TPCC_ALL.*.plan

# ###############################################################
# Helper Targets
# ###############################################################

catalog:        uncatalog
                - db2 -tvf cat-proc.ddl
                - db2 -td$$ -vf cat-func.ddl

uncatalog:
                - db2 -td$$ -vf uncat-func.ddl
                - db2 -tvf uncat-proc.ddl

explain:
                - db2 -tvf $(TPCC_ROOT)$(SLASH)utils$(SLASH)EXPLAIN.DDL

unexplain:
                - db2 -tvf $(TPCC_ROOT)$(SLASH)utils$(SLASH)UNEXPLAIN.DDL

connect:
                - db2 connect to $(TPCC_DBNAME)

disconnect:
                - db2 connect reset
                - db2 terminate

plan:
                - db2exfmt -d $(TPCC_DBNAME) -e $(TPCC_SCHEMA) -s $(TPCC_SCHEMA) -w -1 -n
TPCC_ALL -g -# 0 -o TPCC_ALL.exfmt.plan
                - (export DB2EXPLN_BUFFER=3000000; db2expln -d $(TPCC_DBNAME) -c
$(TPCC_SCHEMA) -p TPCC_ALL -s 0 -g -o TPCC_ALL.expln.plan )

rebind:         connect
                db2 bind tpcc_all_sql.bnd $(BND_OPTS)

# ###############################################################
# Install Targets
# ###############################################################

install:        $(EXE)
                - mkdir $(TPCC_SPDIR)
                $(COPY) ords $(TPCC_SPDIR)
                $(COPY) news $(TPCC_SPDIR)
                $(COPY) dels $(TPCC_SPDIR)

# ###############################################################
# Build Rules
# ###############################################################

.SUFFIXES: $(OBJEXT) .c .sqc

# d230437mte: QUERYOPT 7 required for UNION ALL
# Only stock needs CS , and that can be specified on the SELECT statement
tpcc_all_sql.c:
                @echo "$*.SQC->C:"
                db2 prep $*.sqc $(PRP_OPTS) ISOLATION RR
                db2 bind $*.bnd $(BND_OPTS) QUERYOPT 7

# Stored procedures are built in a special way

tpcc_all_sql$(OBJEXT):
                $(CC) -c tpcc_all_sql.c $(CFLAGS) -D$(TPCC_SPTYPE) $(CFLAGS_OUT)$@

$(EXE): $(UTIL_OBJ) tpcc_all_sql.o
                $(LD_STORP) $(LDFLAGS) $(UTIL_OBJ) tpcc_all_sql.o $(LDFLAGS_OUT)$@

# ###############################################################
# Dependencies
# ###############################################################

# Executables (Stored Procedures)
$(EXE): $(UTIL_OBJ) tpcc_all_sql.o

# Source
tpcc_all_sql$(OBJEXT): tpcc_all_sql.c

# Headers
tpcc_all_sql.c: $(TPCC_ROOT)/include/db2tpcc.h
```

Makefile.rteiface

```
#################################################################################
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#################################################################################

#
# Makefile.rteiface - Makefile for RTE Interface
#

include Makefile.config.rteiface

FILES   = tpcc_all_sql.o tpccdbg.o tpccctx.o
DISTFILES = tpcccli.a lval.h db2tpcc.h

INCLUDE =       -I$(TPCC_SQLLIB)/include -I$(TPCC_ROOT)/include

CFLAGS =        $(CFLAGS_OS) $(CFLAGS_DEBUG) $(INCLUDE) \
            -DSQLA_NOLINES -D$(DB2EDITION) -D$(DB2VERSION) \
            -D$(TPCC_SPTYPE)

all:    copy1 copy2 $(FILES) lib dist

copy1:
        cp $(TPCC_ROOT)/Src.Srv/tpcc_all_sql.c .
        cp $(TPCC_ROOT)/Src.Common/tpccdbg.c .
        cp $(TPCC_ROOT)/Src.Common/tpccctx.c .

copy2:
        cp $(TPCC_ROOT)/include/db2tpcc.h .
```

```
	cp $(TPCC_ROOT)/include/lval.h .

lib:
	ar -q -v tpcccli.a tpcc_all_sql.o tpccdbg.o tpccctx.o

dist:
	tar cvf rteclient.tar tpcccli.a lval.h db2tpcc.h
	cp rteclient.tar $(TPCC_ROOT)/Src.Cli

clean:
	-rm *.h *.c *.o *.a *.tarteclient.tar $(TPCC_ROOT)/Src.Clrteclient.tar $(TPCC_ROOT)/Src.Cli
```

Makefile.config

```
################################################################################
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
################################################################################

#
# Makefile.config - AIX 64-bit
#

# Make Configuration
MAKE=make

# Compiler Configuration.
# DEBUG may be set to "-g", "-DDEBUGIT" "-g -DDEBUGIT" or left blank
CC=xlc
CFLAGS_OS=-qflag=i:i -qlanglvl=ansi -qcpluscmt -DSQLUNIX -DSQLAIX -q64 -O3 -D_LARGE_FILES
CFLAGS_OUT=-o
CFLAGS_DEBUG=

# Linker Configuration
LD_EXEC=xlc
LD_STORP=ld
LDFLAGS_EXEC=-q64
LDFLAGS_STORP=-H512 -T512 -bE:$@.exp -e $@ -lc -b64
LDFLAGS_LIB=-L$(TPCC_SQLLIB)/lib-ldb2
LDFLAGS_OUT=-o

# OS Commands
ERASE=rm -f
ERASEDIR=$(ERASE) -R
MOVE=mv
COPY=cp

# OS File Extensions & Path Separators
OBJEXT=.o
LIBEXT=.a
BINEXT=
SLASH=/
CMDSEP=;
```

tpccdbg.c

```
/******************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
******************************************************************************/

/*
 * tccdbg.c - Debugging Routines
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <time.h>

#include "sqlca.h"
#include "sql.h"
#include "db2tpcc.h"
#include "tpccdbg.h"

#define DEBUG_FILENAME_SZ 128
#define DEBUG_PATH_SIZE 128

void    del_print();
void    new_print();
void    ord_print();
void    pay_print();
void    stk_print();

void current_tmstmp(char *buf);

static int debugInit = 0;
static char debugPath[DEBUG_PATH_SIZE] = "";

/*-------------------------------------------------------------*/
/*       InitializeDebug                                       */
/*-------------------------------------------------------------*/
__inline void InitializeDebug(void) {
  if (debugInit == 0) {
    char *p = getenv("TPCC_DEBUGDIR");
    if (p) {
      strncpy(debugPath, p, DEBUG_PATH_SIZE);
    } else {
      strcpy(debugPath, "/tmp");
    }
    strcat(debugPath, "/");
  }
  debugInit = 1;
}

/*-------------------------------------------------------------*/
/*       sqlerror                                              */
/*-------------------------------------------------------------*/
void sqlerror(int tranType, char *msg, int ptat, SQL_STRUCTURE sqlca *psqlca)
{
  FILE *err_fp = NULL;
  char  err_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];
  char  tranName[16];
  int j,k;
  char  timeStamp[27];
  char  errStr[512] = "";

  InitializeDebug();
  strncpy(err_fn, debugPath, DEBUG_PATH_SIZE);
  current_tmstmp(&timeStamp[0]);
  timeStamp[19] = (char)NULL;
  switch(tranType)
  {
    case NEWORD_SQL:
      // sprintf(err_fn, "%d.err.out", getpid());
      strcat(err_fn, "new.err.out");
      strcpy(tranName, "NEW_ORDER");
      break;

    case DELIVERY_SQL:
      // sprintf(err_fn, "%d.err.out", getpid());
      strcat(err_fn, "del.err.out");
      strcpy(tranName, "DELIVERY");
      break;

    case PAYMENT_SQL:
      // sprintf(err_fn, "%d.err.out", getpid());
      strcat(err_fn, "pay.err.out");
      strcpy(tranName, "PAYMENT");
      break;

    case ORDSTAT_SQL:
      // sprintf(err_fn, "%d.err.out", getpid());
      strcat(err_fn, "ord.err.out");
      strcpy(tranName, "ORDER_STAT");
      break;

    case STOCKLEV_SQL:
      //sprintf(err_fn, "%d.err.out", getpid());
      strcat(err_fn, "stk.err.out");
      strcpy(tranName, "STOCK_LVL");
      break;

    default:
      return;

  }

  /* Generate Formatted Error Message */
  sqlaintp(errStr, 512, 78, psqlca);

  err_fp = fopen(err_fn, "a+");

  fprintf(err_fp, "----------------------------------------\n");
  fprintf(err_fp, "Transaction: %s at point %2d (%s)\n",tranName, ptat, msg);
  fprintf(err_fp, "SQLCODE %d ", psqlca->sqlcode);
  fprintf(err_fp, "PID %d ", getpid());
  fprintf(err_fp, "TIME %s\n", timeStamp);
  fprintf(err_fp, "----------------------------------------\n");
  fprintf(err_fp, "%s", errStr);
  fprintf(err_fp, "----------------------------------------\n");

  if (psqlca->sqlerrmc[0] != ' ' || psqlca->sqlerrmc[1] != ' ')
```

```
  {
    fprintf(err_fp, "slerrmc:  ");

    for(j = 0; j < 5; j++)
    {
      for(k = 0; k < 16; k++) {
        int pos = j * 16 + k;
        if (pos < 70) fprintf(err_fp, "%02x ", psqlca->sqlerrmc[pos]);
        else fprintf(err_fp, "   ");
      }
      fprintf(err_fp,"  |");
      for(k = 0; k < 16; k++) {
        int pos = j * 16 + k;
        char c = ' ';
        if (pos < 70) {
          c = psqlca->sqlerrmc[pos];
          if (!isprint(c)) c = ' ';
        }
        fprintf(err_fp,"%c", c);
      }
      Fprintf(err_fp,"|\n");
      if (j < 4) fprintf(err_fp,"          ");
    }
  }

  fprintf(err_fp, "sqlerrp:  ");
  for(j = 0; j < 8; j++)
    fprintf(err_fp, "%c", psqlca->sqlerrp[j]);
  fprintf(err_fp,"\n");

  fprintf(err_fp, "sqlerrd: ");
  for(j = 0; j < 6; j++)
    fprintf(err_fp, " %d", psqlca->sqlerrd[j]);
  fprintf(err_fp,"\n");

  if (psqlca->sqlwarn[0] != ' ')
  {
    fprintf(err_fp, "sqlwarn: ");
    for(j = 0; j < 8; j++)
      fprintf(err_fp, "%c ", psqlca->sqlwarn[j]);
    fprintf(err_fp,"\n");
  }

  fprintf(err_fp, "\n");

  fclose(err_fp);
}


/*---------------------------------------------------------------*/
/* del_debug                                                     */
/*---------------------------------------------------------------*/
void del_debug (struct out_delivery_struct *delivery_ptr,
                struct in_delivery_struct  *in_delivery,
                char *msg)
{
  char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

  InitializeDebug();
  strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
  strcat(debug_fn, "del.debug.out");
  del_print(delivery_ptr, in_delivery, debug_fn, msg);
}

/*---------------------------------------------------------------*/
/* del_print                                                     */
/*---------------------------------------------------------------*/
void del_print (struct out_delivery_struct *delivery_ptr,
                struct in_delivery_struct  *in_delivery,
                char *filename,
                char *msg)
{
  FILE *debug_fp;
  char  timeStamp[27];
  int   j;

  current_tmstmp(&timeStamp[0]);
  timeStamp[19] = (char)NULL;

  if ((debug_fp = fopen(filename, "a+")) == NULL)
  {
    return;
  }

  fprintf(debug_fp,"Delivery debug information follows %s (%s)\n", timeStamp, msg);
  fprintf(debug_fp, " PID %d ", getpid());
  fprintf(debug_fp,"\n==================================================\n");

  fprintf(debug_fp,"in_delivery_struct {\n");
  fprintf(debug_fp,"\ts_W_ID        = %ld (%lX)\n",
          in_delivery->s_W_ID, in_delivery->s_W_ID);
  fprintf(debug_fp,"\ts_O_CARRIER_ID = %hd (%hX)\n",
          in_delivery->s_O_CARRIER_ID, in_delivery->s_O_CARRIER_ID);
fprintf(debug_fp,"\ts_O_DELIVERY_D = %lld (%llX)\n",
          in_delivery->s_O_DELIVERY_D_time, in_delivery->s_O_DELIVERY_D_time);
  fprintf(debug_fp,"}\n\n");

  fprintf(debug_fp,"out_delivery_struct {\n");
  fprintf(debug_fp,"\ts_transtatus   = %hd (%hX)\n",
          delivery_ptr->s_transtatus,delivery_ptr->s_transtatus);
  fprintf(debug_fp,"\tdeadlocks     = %hd (%hX)\n",
          delivery_ptr->deadlocks,delivery_ptr->deadlocks);

  for (j = 0; j < 10; j++) {
    fprintf(debug_fp,"\ts_O_ID[%d]        = %d\n",
            j, delivery_ptr->s_O_ID[j]);
  }
  fprintf(debug_fp,"\t}\n}\n\n");
  fclose(debug_fp);
}


/*---------------------------------------------------------------*/
/* new_debug                                                     */
/*---------------------------------------------------------------*/
void new_debug (struct out_neword_struct *neword_ptr,
                struct in_neword_struct *in_neword,
                char *msg)
{
  char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

  InitializeDebug();
  strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
  strcat(debug_fn, "new.debug.out");
  new_print(neword_ptr, in_neword, debug_fn, msg);
}

/*---------------------------------------------------------------*/
/* new_print                                                     */
/*---------------------------------------------------------------*/
void new_print (struct out_neword_struct *neword_ptr,
                struct in_neword_struct *in_neword,
                char *filename,
                char *msg)
{
  FILE *debug_fp;
  char  timeStamp[27];
  int   j, items;

  current_tmstmp(&timeStamp[0]);
  timeStamp[19] = (char)NULL;

  if ((debug_fp = fopen(filename, "a+")) == NULL)
  {
    return;
  }

  fprintf(debug_fp,"New order debug information follows %s (%s)\n", timeStamp, msg);
  fprintf(debug_fp, " PID %d ", getpid());
  fprintf(debug_fp,"\n==================================================\n");

  fprintf(debug_fp,"in_neword_struct {\n");

  fprintf(debug_fp,"\ts_C_ID        = %ld (%lX)\n",
          in_neword->s_C_ID, in_neword->s_C_ID);
  fprintf(debug_fp,"\ts_W_ID        = %ld (%lX)\n",
          in_neword->s_W_ID, in_neword->s_W_ID);
  fprintf(debug_fp,"\ts_D_ID        = %hd (%hX)\n",
          in_neword->s_D_ID, in_neword->s_D_ID);
  fprintf(debug_fp,"\ts_O_OL_CNT    = %hd (%hX)\n",
          in_neword->s_O_OL_CNT, in_neword->s_O_OL_CNT);
  fprintf(debug_fp,"\ts_all_local   = %hd (%hX)\n",
          in_neword->s_all_local, in_neword->s_all_local);
  fprintf(debug_fp,"\ts_O_ENTRY_D   = %lld (%llX)\n",
          in_neword->s_O_ENTRY_D_time, in_neword->s_O_ENTRY_D_time);
// fprintf(debug_fp,"\ts_transtatus  = %hd (%hX)\n",
//         in_neword->s_transtatus, in_neword->s_transtatus);
// fprintf(debug_fp,"\tduplicate_items= %hd (%hX)\n",
//         in_neword->duplicate_items, in_neword->duplicate_items);

  fprintf(debug_fp,"\titems {\n");
  items = in_neword->s_O_OL_CNT;
  for (j=0; j<items; j++) {
    if(j != 0)
      fprintf(debug_fp,"\n");
    fprintf(debug_fp,"\t\ts_OL_I_ID[%d]       = %ld (%lX)\n",
            j, in_neword->in_item[j].s_OL_I_ID, in_neword->in_item[j].s_OL_I_ID);
    fprintf(debug_fp,"\t\ts_OL_SUPPLY_W_ID[%d]= %ld (%lX)\n",
            j, in_neword->in_item[j].s_OL_SUPPLY_W_ID, in_neword->in_item[j].s_OL_SUPPLY_W_ID);
    fprintf(debug_fp,"\t\ts_OL_QUANTITY[%d]   = %hd (%hX)\n",
            j, in_neword->in_item[j].s_OL_QUANTITY, in_neword->in_item[j].s_OL_QUANTITY);
  }
  fprintf(debug_fp,"\t}\n}\n\n");

  fprintf(debug_fp,"out_neword_struct {\n");
  fprintf(debug_fp,"\ts_C_LAST      = %s\n",
          neword_ptr->s_C_LAST);
  fprintf(debug_fp,"\ts_C_CREDIT    = %s\n",
          neword_ptr->s_C_CREDIT);
  fprintf(debug_fp,"\ts_W_TAX       = %d\n",
          neword_ptr->s_W_TAX);
  fprintf(debug_fp,"\ts_D_TAX       = %d\n",
          neword_ptr->s_D_TAX);
  fprintf(debug_fp,"\ts_C_DISCOUNT  = %d\n",
          neword_ptr->s_C_DISCOUNT);
  fprintf(debug_fp,"\ts_O_ID        = %ld (%lX)\n",
          neword_ptr->s_O_ID, neword_ptr->s_O_ID);
  fprintf(debug_fp,"\ts_O_OL_CNT    = %hd (%hX)\n",
```

```
        neword_ptr->s_O_OL_CNT, neword_ptr->s_O_OL_CNT);
  fprintf(debug_fp,"\ts_O_ENTRY_D   = %lld (%llX)\n",
        in_neword->s_O_ENTRY_D_time, neword_ptr->s_O_ENTRY_D_time);
  fprintf(debug_fp,"\ts_total_amount = %ld\n",
        neword_ptr->s_total_amount);
  fprintf(debug_fp,"\ts_transtatus  = %hd (%hX)\n",
        neword_ptr->s_transtatus, neword_ptr->s_transtatus);
  fprintf(debug_fp,"\tdeadlocks     = %hd (%hX)\n",
        neword_ptr->deadlocks, neword_ptr->deadlocks);

// fprintf(debug_fp,"\ts_W_ID        = %ld (%lX)\n",
//         neword_ptr->s_W_ID, neword_ptr->s_W_ID);
// fprintf(debug_fp,"\ts_D_ID        = %hd (%hX)\n",
//         neword_ptr->s_D_ID, neword_ptr->s_D_ID);
// fprintf(debug_fp,"\ts_all_local   = %hd (%hX)\n",
//         neword_ptr->s_all_local, neword_ptr->s_all_local);
// fprintf(debug_fp,"\tduplicate_items= %hd (%hX)\n",
//         neword_ptr->duplicate_items, neword_ptr->duplicate_items);

  fprintf(debug_fp,"\titems {\n");
  items = neword_ptr->s_O_OL_CNT;
  for (j=0; j<items; j++) {
    if(j != 0)
      fprintf(debug_fp,"\n");
    fprintf(debug_fp,"\t\ts_I_NAME[%d]        = %s\n",
          j, neword_ptr->item[j].s_I_NAME);
    fprintf(debug_fp,"\t\ts_I_PRICE[%d]       = %d\n",
          j, neword_ptr->item[j].s_I_PRICE);
    fprintf(debug_fp,"\t\ts_OL_AMOUNT[%d]     = %ld\n",
          j, neword_ptr->item[j].s_OL_AMOUNT);
    fprintf(debug_fp,"\t\ts_S_QUANTITY[%d]    = %hd (%hX)\n",
          j, neword_ptr->item[j].s_S_QUANTITY, neword_ptr->item[j].s_S_QUANTITY);
    fprintf(debug_fp,"\t\ts_brand_generic[%d] = %c\n",
          j, neword_ptr->item[j].s_brand_generic);
  }
  fprintf(debug_fp,"\t}\n}\n\n");
  fclose(debug_fp);
}



/*----------------------------------------------------------------*/
/* ord_debug                                                      */
/*----------------------------------------------------------------*/
void ord_debug (struct out_ordstat_struct *ordstat_ptr,
            struct in_ordstat_struct  *in_ordstat,
            char *msg)
{
  char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

  InitializeDebug();
  strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
  strcat(debug_fn, "ord.debug.out");
  ord_print(ordstat_ptr, in_ordstat, debug_fn, msg);
}

/*----------------------------------------------------------------*/
/* ord_print                                                      */
/*----------------------------------------------------------------*/
void ord_print (struct out_ordstat_struct *ordstat_ptr,
            struct in_ordstat_struct  *in_ordstat,
            char *filename,
            char *msg)
{
  FILE *debug_fp;
  char  timeStamp[27];
  int   j, items;

  current_tmstmp(&timeStamp[0]);
  timeStamp[19] = (char)NULL;

  if ((debug_fp = fopen(filename, "a+")) == NULL)
  {
    return;
  }

  fprintf(debug_fp,"Order status debug information follows %s (%s)\n", timeStamp, msg);
  fprintf(debug_fp, " PID %d ", getpid());
  fprintf(debug_fp,"\n==========================================================\n");

  fprintf(debug_fp,"in_ordstat_struct {\n");
  fprintf(debug_fp,"\ts_W_ID        = %ld (%lX)\n",
        in_ordstat->s_W_ID, in_ordstat->s_W_ID);
  fprintf(debug_fp,"\ts_D_ID        = %hd (%hX)\n",
        in_ordstat->s_D_ID, in_ordstat->s_D_ID);
  fprintf(debug_fp,"\ts_C_ID        = %ld (%lX)\n",
        in_ordstat->s_C_ID, in_ordstat->s_C_ID);
  fprintf(debug_fp,"\ts_C_LAST      = %s\n",
        in_ordstat->s_C_LAST);
  fprintf(debug_fp,"}\n\n");

  fprintf(debug_fp,"out_ordstat_struct {\n");
  fprintf(debug_fp,"\ts_C_ID        = %ld (%lX)\n",
        ordstat_ptr->s_C_ID, ordstat_ptr->s_C_ID);
  fprintf(debug_fp,"\ts_C_FIRST     = %s\n",
        ordstat_ptr->s_C_FIRST);
  fprintf(debug_fp,"\ts_C_MIDDLE    = %s\n",
        ordstat_ptr->s_C_MIDDLE);
  fprintf(debug_fp,"\ts_C_LAST      = %s\n",
        ordstat_ptr->s_C_LAST);
  fprintf(debug_fp,"\ts_C_BALANCE   = %lld\n",
        ordstat_ptr->s_C_BALANCE);
  fprintf(debug_fp,"\ts_O_ID        = %ld (%lX)\n",
        ordstat_ptr->s_O_ID, ordstat_ptr->s_O_ID);
  fprintf(debug_fp,"\ts_O_ENTRY_D   = %lld (%llX)\n",
        ordstat_ptr->s_O_ENTRY_D_time, ordstat_ptr->s_O_ENTRY_D_time);
  fprintf(debug_fp,"\ts_O_CARRIER_ID= %hd (%hX)\n",
        ordstat_ptr->s_O_CARRIER_ID, ordstat_ptr->s_O_CARRIER_ID);
  fprintf(debug_fp,"\ts_ol_cnt      = %hd (%hX)\n",
        ordstat_ptr->s_ol_cnt, ordstat_ptr->s_ol_cnt);
  fprintf(debug_fp,"\ts_transtatus  = %hd (%hX)\n",
        ordstat_ptr->s_transtatus, ordstat_ptr->s_transtatus);
  fprintf(debug_fp,"\tdeadlocks     = %hd (%hX)\n",
        ordstat_ptr->deadlocks, ordstat_ptr->deadlocks);

  fprintf(debug_fp,"\titems {\n");
  items = ordstat_ptr->s_ol_cnt;
  for (j = 0; j < items; j++) {
    if(j != 0)
      fprintf(debug_fp,"\n");
    fprintf(debug_fp,"\t\ts_OL_SUPPLY_W_ID[%d]= %ld (%lX)\n",
          j, ordstat_ptr->item[j].s_OL_SUPPLY_W_ID, ordstat_ptr->item[j].s_OL_SUPPLY_W_ID);
    fprintf(debug_fp,"\t\ts_OL_I_ID[%d]       = %ld (%lX)\n",
          j, ordstat_ptr->item[j].s_OL_I_ID, ordstat_ptr->item[j].s_OL_I_ID);
    fprintf(debug_fp,"\t\ts_OL_QUANTITY[%d]   = %hd (%hX)\n",
          j, ordstat_ptr->item[j].s_OL_QUANTITY, ordstat_ptr->item[j].s_OL_QUANTITY);
    fprintf(debug_fp,"\t\ts_OL_AMOUNT[%d]     = %d\n",
          j, ordstat_ptr->item[j].s_OL_AMOUNT);
    fprintf(debug_fp,"\t\ts_OL_DELIVERY_D[%d] = %lld (%llX)\n",
          j,ordstat_ptr->item[j].s_OL_DELIVERY_D_time,
          ordstat_ptr->item[j].s_OL_DELIVERY_D_time);
  }
  fprintf(debug_fp,"\t}\n}\n\n");
  fclose(debug_fp);
}


/*----------------------------------------------------------------*/
/* pay_debug                                                      */
/*----------------------------------------------------------------*/
void pay_debug (struct out_payment_struct *payment_ptr,
            struct in_payment_struct  *in_payment,
            char *msg)
{
  char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

  InitializeDebug();
  strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
  strcat(debug_fn, "pay.debug.out");
  pay_print(payment_ptr, in_payment, debug_fn, msg);
}

/*----------------------------------------------------------------*/
/* pay_print                                                      */
/*----------------------------------------------------------------*/
void pay_print (struct out_payment_struct *payment_ptr,
            struct in_payment_struct  *in_payment,
            char *filename,
            char *msg)
{
  FILE *debug_fp;
  char  timeStamp[27];

  current_tmstmp(&timeStamp[0]);
  timeStamp[19] = (char)NULL;

  if ((debug_fp = fopen(filename, "a+")) == NULL)
  {
    return;
  }

  fprintf(debug_fp,"Payment debug information follows %s (%s)\n", timeStamp, msg);
  fprintf(debug_fp, " PID %d ", getpid());
  fprintf(debug_fp,"\n==========================================================\n");

  fprintf(debug_fp,"in_payment_struct {\n");
  fprintf(debug_fp,"\ts_H_AMOUNT    = %lld (%llX)\n",
        in_payment->s_H_AMOUNT, in_payment->s_H_AMOUNT);
  fprintf(debug_fp,"\ts_C_ID        = %ld (%lX)\n",
        in_payment->s_C_ID, in_payment->s_C_ID);
  fprintf(debug_fp,"\ts_W_ID        = %ld (%lX)\n",
        in_payment->s_W_ID, in_payment->s_W_ID);
  fprintf(debug_fp,"\ts_D_ID        = %hd (%hX)\n",
        in_payment->s_D_ID, in_payment->s_D_ID);
  fprintf(debug_fp,"\ts_C_D_ID      = %hd (%hX)\n",
        in_payment->s_C_D_ID, in_payment->s_C_D_ID);
  fprintf(debug_fp,"\ts_C_W_ID      = %ld (%lX)\n",
        in_payment->s_C_W_ID, in_payment->s_C_W_ID);
  fprintf(debug_fp,"\ts_C_LAST      = %s\n",
        in_payment->s_C_LAST);
  fprintf(debug_fp,"\ts_H_DATE      = %lld (%llX)\n",
        in_payment->s_H_DATE_time, in_payment->s_H_DATE_time);
  fprintf(debug_fp,"\n}\n\n");

  fprintf(debug_fp,"out_payment_struct {\n");
  fprintf(debug_fp,"\ts_H_DATE      = %lld (%llX)\n",
        in_payment->s_H_DATE_time, in_payment->s_H_DATE_time);
```

```
  fprintf(debug_fp,"\ts_C_CREDIT_LIM = %lld\n",
        payment_ptr->s_C_CREDIT_LIM);
  fprintf(debug_fp,"\ts_C_DISCOUNT  = %d\n",
        payment_ptr->s_C_DISCOUNT);
  fprintf(debug_fp,"\ts_C_BALANCE   = %lld\n",
        payment_ptr->s_C_BALANCE);
  fprintf(debug_fp,"\ts_C_ID        = %ld (%lX)\n",
        payment_ptr->s_C_ID, payment_ptr->s_C_ID);
  fprintf(debug_fp,"\ts_W_STREET_1  = %s\n",
        payment_ptr->s_W_STREET_1);
  fprintf(debug_fp,"\ts_W_STREET_2  = %s\n",
        payment_ptr->s_W_STREET_2);
  fprintf(debug_fp,"\ts_W_CITY      = %s\n",
        payment_ptr->s_W_CITY);
  fprintf(debug_fp,"\ts_W_STATE     = %s\n",
        payment_ptr->s_W_STATE);
  fprintf(debug_fp,"\ts_W_ZIP       = %s\n",
        payment_ptr->s_W_ZIP);
  fprintf(debug_fp,"\ts_D_STREET_1  = %s\n",
        payment_ptr->s_D_STREET_1);
  fprintf(debug_fp,"\ts_D_STREET_2  = %s\n",
        payment_ptr->s_D_STREET_2);
  fprintf(debug_fp,"\ts_D_CITY      = %s\n",
        payment_ptr->s_D_CITY);
  fprintf(debug_fp,"\ts_D_STATE     = %s\n",
        payment_ptr->s_D_STATE);
  fprintf(debug_fp,"\ts_D_ZIP       = %s\n",
        payment_ptr->s_D_ZIP);
  fprintf(debug_fp,"\ts_C_FIRST     = %s\n",
        payment_ptr->s_C_FIRST);
  fprintf(debug_fp,"\ts_C_MIDDLE    = %s\n",
        payment_ptr->s_C_MIDDLE);
  fprintf(debug_fp,"\ts_C_LAST      = %s\n",
        payment_ptr->s_C_LAST);
  fprintf(debug_fp,"\ts_C_STREET_1  = %s\n",
        payment_ptr->s_C_STREET_1);
  fprintf(debug_fp,"\ts_C_STREET_2  = %s\n",
        payment_ptr->s_C_STREET_2);
  fprintf(debug_fp,"\ts_C_CITY      = %s\n",
        payment_ptr->s_C_CITY);
  fprintf(debug_fp,"\ts_C_STATE     = %s\n",
        payment_ptr->s_C_STATE);
  fprintf(debug_fp,"\ts_C_ZIP       = %s\n",
        payment_ptr->s_C_ZIP);
  fprintf(debug_fp,"\ts_C_PHONE     = %s\n",
        payment_ptr->s_C_PHONE);
  fprintf(debug_fp,"\ts_C_SINCE     = %lld (%llX)\n",
        payment_ptr->s_C_SINCE_time, payment_ptr->s_C_SINCE_time);
  fprintf(debug_fp,"\ts_C_CREDIT    = %s\n",
        payment_ptr->s_C_CREDIT);
  fprintf(debug_fp,"\ts_C_DATA      = %s\n",
        payment_ptr->s_C_DATA);
  fprintf(debug_fp,"\ts_transtatus  = %hd (%hX)\n",
        payment_ptr->s_transtatus,payment_ptr->s_transtatus);
  fprintf(debug_fp,"\tdeadlocks     = %hd (%hX)\n",
        payment_ptr->deadlocks,payment_ptr->deadlocks);
  fprintf(debug_fp,"\n}\n\n");
  fclose(debug_fp);
}


/*-------------------------------------------------------------*/
/* stk_debug                                                   */
/*-------------------------------------------------------------*/
void stk_debug (struct out_stocklev_struct *stocklev,
            struct in_stocklev_struct  *in_stocklev,
            char *msg)
{
  char debug_fn[DEBUG_PATH_SIZE + DEBUG_FILENAME_SZ];

  InitializeDebug();
  strncpy(debug_fn, debugPath, DEBUG_PATH_SIZE);
  strcat(debug_fn, "stk.debug.out");
  stk_print(stocklev, in_stocklev, debug_fn, msg);
}

/*-------------------------------------------------------------*/
/* stk_print                                                   */
/*-------------------------------------------------------------*/
void stk_print (struct out_stocklev_struct *stocklev,
            struct in_stocklev_struct  *in_stocklev,
            char *filename,
            char *msg)
{
  FILE *debug_fp;
  char  timeStamp[27];

  current_tmstmp(&timeStamp[0]);
  timeStamp[19] = (char)NULL;

  if ((debug_fp = fopen(filename, "a+")) == NULL)
  {
    return;
  }

  fprintf(debug_fp,"Stock level debug information follows %s (%s)\n", timeStamp, msg);
  fprintf(debug_fp, " PID %d ", getpid());
  fprintf(debug_fp,"\n==========================================================\n");
  fprintf(debug_fp,"in_stocklev_struct {\n");
  fprintf(debug_fp,"\ts_W_ID        = %ld (%lX)\n",
        in_stocklev->s_W_ID, in_stocklev->s_W_ID);
  fprintf(debug_fp,"\ts_D_ID        = %hd (%hX)\n",
        in_stocklev->s_D_ID, in_stocklev->s_D_ID);
  fprintf(debug_fp,"\ts_threshold   = %hd (%hX)\n",
        in_stocklev->s_threshold, in_stocklev->s_threshold);
  fprintf(debug_fp,"}\n\n");

  fprintf(debug_fp,"out_stocklev_struct {\n");
  fprintf(debug_fp,"\ts_transtatus  = %hd (%hX)\n",
        stocklev->s_transtatus, stocklev->s_transtatus);
  fprintf(debug_fp,"\tdeadlocks     = %hd (%hX)\n",
        stocklev->deadlocks, stocklev->deadlocks);
  fprintf(debug_fp,"\ts_low_stock   = %ld (%lX)\n",
        stocklev->s_low_stock, stocklev->s_low_stock);
  fprintf(debug_fp,"}\n\n");
  fclose(debug_fp);
}

void current_tmstmp(char *buf)
{
  time_t t = time(NULL);
  strncpy(buf,ctime(&t),19);
}
```

Tpccmisc.c

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * tpccmisc.c - Miscellaneous routines
 */

#include <sys/types.h>
#include <sys/time.h>

double current_time(void)
{
  return(time(NULL));

}
```

cat-func.ddl

```
-------------------------------------------------------------------------
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
-------------------------------------------------------------------------
--
-- cat-func.ddl - Create table functions
--

--
-- DELIVERY
--

CREATE FUNCTION DEL(   W_ID          INTEGER
                     , D_ID          SMALLINT
                     , CARRIER_ID    SMALLINT
                     , DELIVERY_D    BIGINT
                   )

RETURNS TABLE ( O_ID INTEGER )

SPECIFIC DELIVERY

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

  DECLARE O_ID   INTEGER ;
  DECLARE C_ID   INTEGER ;
```

```
 DECLARE AMOUNT INTEGER ;

 SET VAR.O_ID = ( SELECT NO_O_ID

                FROM OLD TABLE ( DELETE

                                 FROM ( SELECT NO_O_ID

                                        FROM NEW_ORDER

                                        WHERE NO_W_ID = DEL.W_ID
                                         AND NO_D_ID = DEL.D_ID

                                        ORDER BY NO_O_ID ASC

                                        FETCH FIRST 1 ROW ONLY
                                      ) AS NEW_ORDER
                               ) AS D
            )
 ;

 SET VAR.C_ID = ( SELECT O_C_ID

                FROM OLD TABLE ( UPDATE ORDERS

                                 SET O_CARRIER_ID = DEL.CARRIER_ID

                                WHERE O_W_ID = DEL.W_ID
                                 AND O_D_ID = DEL.D_ID
                                 AND O_ID   = VAR.O_ID
                               ) AS U
            )
 ;

 SET VAR.AMOUNT = ( SELECT SUM( OL_AMOUNT )

                  FROM OLD TABLE ( UPDATE ORDER_LINE

                                   SET OL_DELIVERY_D = DEL.DELIVERY_D

                                  WHERE OL_W_ID = DEL.W_ID
                                   AND OL_D_ID = DEL.D_ID
                                   AND OL_O_ID = VAR.O_ID
                                 ) AS U
             )
 ;

 UPDATE CUSTOMER

    SET   C_BALANCE     = C_BALANCE + VAR.AMOUNT
        , C_DELIVERY_CNT = C_DELIVERY_CNT + SMALLINT( 1 )

  WHERE C_W_ID = DEL.W_ID
    AND C_D_ID = DEL.D_ID
    AND C_ID   = VAR.C_ID
 ;

 RETURN VALUES VAR.O_ID ;

END
$


--
-- ORDER STATUS
--

CREATE FUNCTION ORD_C_LAST(   W_ID INTEGER
                    , D_ID SMALLINT
                    , C_LAST VARCHAR(16)
                   )

RETURNS TABLE(   O_ID        INTEGER
           , O_CARRIER_ID SMALLINT
           , O_ENTRY_D    BIGINT
           , C_BALANCE    BIGINT
           , C_FIRST      VARCHAR(16)
           , C_MIDDLE     CHAR(2)
           , C_ID         INTEGER
          )

SPECIFIC ORD_C_LAST

MODIFIES SQL DATA NO EXTERNAL ACTION DETERMINISTIC LANGUAGE SQL

VAR: BEGIN ATOMIC

 DECLARE C_BALANCE    BIGINT ;
 DECLARE C_FIRST      VARCHAR(16) ;
 DECLARE C_MIDDLE     CHAR(2) ;
 DECLARE C_ID         INTEGER ;
 DECLARE O_ID         INTEGER;
 DECLARE O_CARRIER_ID SMALLINT;
 DECLARE O_ENTRY_D    BIGINT;

 SET ( C_BALANCE, C_FIRST, C_MIDDLE, C_ID )

   = ( SELECT  C_BALANCE, C_FIRST, C_MIDDLE , C_ID
         FROM ( SELECT   C_ID
                      , C_BALANCE
                      , C_FIRST
                      , C_MIDDLE
                      , COUNT(*) OVER() AS COUNT
                      , ROWNUMBER() OVER (ORDER BY C_FIRST) AS NUM

                 FROM CUSTOMER
                WHERE C_W_ID = ORD_C_LAST.W_ID
                 AND C_D_ID = ORD_C_LAST.D_ID
                 AND C_LAST = ORD_C_LAST.C_LAST

              ) AS V1

        WHERE NUM = (COUNT + BIGINT( 1 ) ) / BIGINT( 2 )
     )
 ;

 SET ( O_ID , O_CARRIER_ID , O_ENTRY_D )
   =   ( SELECT   O_ID
               , O_CARRIER_ID
               , O_ENTRY_D

           FROM ORDERS

          WHERE O_W_ID = ORD_C_LAST.W_ID
           AND O_D_ID = ORD_C_LAST.D_ID
           AND O_C_ID = VAR.C_ID

          ORDER BY O_ID DESC
          FETCH FIRST 1 ROW ONLY

       )
 ;


 RETURN VALUES (   VAR.O_ID
               , VAR.O_CARRIER_ID
               , VAR.O_ENTRY_D
               , VAR.C_BALANCE
               , VAR.C_FIRST
               , VAR.C_MIDDLE
               , VAR.C_ID
              )
 ;


END $

CREATE FUNCTION ORD_C_ID(   W_ID INTEGER
                    , D_ID SMALLINT
                    , C_ID INTEGER
                   )

RETURNS TABLE(   O_ID        INTEGER
           , O_CARRIER_ID SMALLINT
           , O_ENTRY_D    BIGINT
           , C_BALANCE    BIGINT
           , C_FIRST      VARCHAR(16)
           , C_MIDDLE     CHAR(2)
           , C_LAST       VARCHAR(16)
          )

SPECIFIC ORD_C_ID

MODIFIES SQL DATA NO EXTERNAL ACTION DETERMINISTIC LANGUAGE SQL

VAR: BEGIN ATOMIC

 DECLARE C_BALANCE    BIGINT ;
 DECLARE C_FIRST      VARCHAR(16) ;
 DECLARE C_MIDDLE     CHAR(2) ;
 DECLARE C_LAST       VARCHAR(16) ;
 DECLARE O_ID         INTEGER;
 DECLARE O_CARRIER_ID SMALLINT;
 DECLARE O_ENTRY_D    BIGINT;

 SET ( C_BALANCE, C_FIRST, C_MIDDLE, C_LAST )

  = ( SELECT  C_BALANCE, C_FIRST, C_MIDDLE, C_LAST

        FROM CUSTOMER

       WHERE C_ID   = ORD_C_ID.C_ID
        AND C_W_ID = ORD_C_ID.W_ID
        AND C_D_ID = ORD_C_ID.D_ID
    )
 ;

 SET (O_ID, O_CARRIER_ID, O_ENTRY_D)

  =   ( SELECT   O_ID
               , O_CARRIER_ID
               , O_ENTRY_D

          FROM ORDERS

         WHERE O_W_ID = ORD_C_ID.W_ID
          AND O_D_ID = ORD_C_ID.D_ID
```

```
            AND O_C_ID = ORD_C_ID.C_ID

        ORDER BY O_ID DESC
        FETCH FIRST 1 ROW ONLY
      )
 ;

 RETURN VALUES (   VAR.O_ID
             , VAR.O_CARRIER_ID
             , VAR.O_ENTRY_D
             , VAR.C_BALANCE
             , VAR.C_FIRST
             , VAR.C_MIDDLE
             , VAR.C_LAST
           ) ;

END $



--
-- PAYMENT
--

CREATE FUNCTION PAY_C_LAST(  W_ID INTEGER
                   , D_ID SMALLINT
                   , C_W_ID INTEGER
                   , C_D_ID SMALLINT
                   , C_LAST VARCHAR(16)
                   , H_DATE BIGINT
                   , H_AMOUNT BIGINT
                   , BAD_CREDIT_PREFIX VARCHAR(28)
                  )

RETURNS TABLE(   W_STREET_1 CHAR(20)
           , W_STREET_2 CHAR(20)
           , W_CITY CHAR(20)
           , W_STATE CHAR(2)
           , W_ZIP CHAR(9)
           , D_STREET_1 CHAR(20)
           , D_STREET_2 CHAR(20)
           , D_CITY CHAR(20)
           , D_STATE CHAR(2)
           , D_ZIP CHAR(9)
           , C_ID INTEGER
           , C_FIRST VARCHAR(16)
           , C_MIDDLE CHAR(2)
           , C_STREET_1 VARCHAR(20)
           , C_STREET_2 VARCHAR(20)
           , C_CITY VARCHAR(20)
           , C_STATE CHAR(2)
           , C_ZIP CHAR(9)
           , C_PHONE CHAR(16)
           , C_SINCE BIGINT
           , C_CREDIT CHAR(2)
           , C_CREDIT_LIM BIGINT
           , C_DISCOUNT INTEGER
           , C_BALANCE BIGINT
           , C_DATA CHAR(200)
         )

SPECIFIC PAY_C_LAST

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

  DECLARE  W_NAME CHAR(10) ;
  DECLARE  D_NAME CHAR(10) ;

  DECLARE  W_STREET_1 CHAR(20) ;
  DECLARE  W_STREET_2 CHAR(20) ;
  DECLARE  W_CITY CHAR(20) ;
  DECLARE  W_STATE CHAR(2) ;
  DECLARE  W_ZIP CHAR(9) ;

  DECLARE  D_STREET_1 CHAR(20) ;
  DECLARE  D_STREET_2 CHAR(20) ;
  DECLARE  D_CITY CHAR(20) ;
  DECLARE  D_STATE CHAR(2) ;
  DECLARE  D_ZIP CHAR(9) ;

  DECLARE  C_ID INTEGER ;

  DECLARE  C_FIRST VARCHAR(16) ;
  DECLARE  C_MIDDLE CHAR(2) ;
  DECLARE  C_STREET_1 VARCHAR(20) ;
  DECLARE  C_STREET_2 VARCHAR(20) ;
  DECLARE  C_CITY VARCHAR(20) ;
  DECLARE  C_STATE CHAR(2) ;
  DECLARE  C_ZIP CHAR(9) ;
  DECLARE  C_PHONE CHAR(16) ;
  DECLARE  C_SINCE BIGINT ;
  DECLARE  C_CREDIT CHAR(2) ;
  DECLARE  C_CREDIT_LIM BIGINT ;
  DECLARE  C_DISCOUNT INTEGER ;
  DECLARE  C_BALANCE BIGINT ;
  DECLARE  C_DATA CHAR(200) ;

  SET ( D_NAME, D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP)

   =  ( SELECT   D_NAME, D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP

         FROM OLD TABLE ( UPDATE DISTRICT

                     SET D_YTD = D_YTD + PAY_C_LAST.H_AMOUNT

                   WHERE D_W_ID = PAY_C_LAST.W_ID
                     AND D_ID   = PAY_C_LAST.D_ID
                  ) AS U
     )
  ;

  SET ( C_ID )

   = ( SELECT C_ID
       FROM ( SELECT   C_ID
                 , COUNT(*) OVER() AS COUNT
                 , ROWNUMBER() OVER (ORDER BY C_FIRST) AS NUM

              FROM CUSTOMER
              WHERE C_LAST = PAY_C_LAST.C_LAST
                AND C_W_ID = PAY_C_LAST.C_W_ID
                AND C_D_ID = PAY_C_LAST.C_D_ID
            ) AS T

      WHERE NUM = (COUNT + BIGINT( 1 ) ) / BIGINT( 2 )
     )
  ;


  SET (   C_ID, C_FIRST, C_MIDDLE, 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 )

   = ( SELECT   C_ID, C_FIRST, C_MIDDLE, 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
           , CASE WHEN C_CREDIT = 'BC' THEN SUBSTR(C_DATA, 1, 200) ELSE NULL END AS
C_DATA

         FROM NEW TABLE ( UPDATE CUSTOMER

                     SET   C_BALANCE    = C_BALANCE - PAY_C_LAST.H_AMOUNT
                       , C_YTD_PAYMENT = C_YTD_PAYMENT + PAY_C_LAST.H_AMOUNT
                       , C_PAYMENT_CNT = C_PAYMENT_CNT + SMALLINT( 1 )

                       , C_DATA = CASE WHEN C_CREDIT = 'BC'
                                  THEN    CHAR( C_ID )           -- 11 bytes long
                                     || BAD_CREDIT_PREFIX       -- 28 bytes long
                                     || SUBSTR( C_DATA, 1, 461 ) -- 461 + 39 = 500
                                  ELSE C_DATA
                                END

                   WHERE C_W_ID = PAY_C_LAST.C_W_ID
                     AND C_D_ID = PAY_C_LAST.C_D_ID
                     AND C_ID   = VAR.C_ID
                  ) AS U
     )
  ;

  SET ( W_NAME, W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP )

   = ( SELECT W_NAME, W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP

         FROM OLD TABLE ( UPDATE WAREHOUSE

                     SET W_YTD = W_YTD + PAY_C_LAST.H_AMOUNT

                   WHERE W_ID = PAY_C_LAST.W_ID
                  ) AS U
     )
  ;

  INSERT

    INTO HISTORY ( H_C_ID, H_C_D_ID, H_C_W_ID, H_D_ID, H_W_ID, H_DATA, H_DATE,
H_AMOUNT )

    VALUES (   VAR.C_ID
           , PAY_C_LAST.C_D_ID
           , PAY_C_LAST.C_W_ID
           , PAY_C_LAST.D_ID
           , PAY_C_LAST.W_ID
           , VAR.W_NAME || CHAR( '    ', 4 )  || VAR.D_NAME
           , PAY_C_LAST.H_DATE
           , PAY_C_LAST.H_AMOUNT
         )
  ;

  RETURN VALUES (   W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP
              , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
              , C_ID, C_FIRST, C_MIDDLE, 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
            )
  ;
```

```
END
$

CREATE FUNCTION PAY_C_ID(  W_ID INTEGER
                , D_ID SMALLINT
                , C_W_ID INTEGER
                , C_D_ID SMALLINT
                , C_ID INTEGER
                , H_DATE BIGINT
                , H_AMOUNT BIGINT
                , BAD_CREDIT_PREFIX VARCHAR(34)
               )

RETURNS TABLE(   W_STREET_1 CHAR(20)
           , W_STREET_2 CHAR(20)
           , W_CITY CHAR(20)
           , W_STATE CHAR(2)
           , W_ZIP CHAR(9)
           , D_STREET_1 CHAR(20)
           , D_STREET_2 CHAR(20)
           , D_CITY CHAR(20)
           , D_STATE CHAR(2)
           , D_ZIP CHAR(9)
           , C_LAST VARCHAR(16)
           , C_FIRST VARCHAR(16)
           , C_MIDDLE CHAR(2)
           , C_STREET_1 VARCHAR(20)
           , C_STREET_2 VARCHAR(20)
           , C_CITY VARCHAR(20)
           , C_STATE CHAR(2)
           , C_ZIP CHAR(9)
           , C_PHONE CHAR(16)
           , C_SINCE BIGINT
           , C_CREDIT CHAR(2)
           , C_CREDIT_LIM BIGINT
           , C_DISCOUNT INTEGER
           , C_BALANCE BIGINT
           , C_DATA CHAR(200)
         )
SPECIFIC PAY_C_ID

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

 DECLARE  W_NAME CHAR(10) ;
 DECLARE  D_NAME CHAR(10) ;

 DECLARE  W_STREET_1 CHAR(20) ;
 DECLARE  W_STREET_2 CHAR(20) ;
 DECLARE  W_CITY CHAR(20) ;
 DECLARE  W_STATE CHAR(2) ;
 DECLARE  W_ZIP CHAR(9) ;

 DECLARE  D_STREET_1 CHAR(20) ;
 DECLARE  D_STREET_2 CHAR(20) ;
 DECLARE  D_CITY CHAR(20) ;
 DECLARE  D_STATE CHAR(2) ;
 DECLARE  D_ZIP CHAR(9) ;

 DECLARE  C_LAST VARCHAR(16) ;

 DECLARE  C_FIRST VARCHAR(16) ;
 DECLARE  C_MIDDLE CHAR(2) ;
 DECLARE  C_STREET_1 VARCHAR(20) ;
 DECLARE  C_STREET_2 VARCHAR(20) ;
 DECLARE  C_CITY VARCHAR(20) ;
 DECLARE  C_STATE CHAR(2) ;
 DECLARE  C_ZIP CHAR(9) ;
 DECLARE  C_PHONE CHAR(16) ;
 DECLARE  C_SINCE BIGINT ;
 DECLARE  C_CREDIT CHAR(2) ;
 DECLARE  C_CREDIT_LIM BIGINT ;
 DECLARE  C_DISCOUNT INTEGER ;
 DECLARE  C_BALANCE BIGINT ;
 DECLARE  C_DATA CHAR(200) ;


 SET ( D_NAME, D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP )

  =  ( SELECT   D_NAME, D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP

       FROM OLD TABLE ( UPDATE DISTRICT

                   SET D_YTD = D_YTD + PAY_C_ID.H_AMOUNT

                  WHERE D_W_ID = PAY_C_ID.W_ID
                    AND D_ID   = PAY_C_ID.D_ID
                ) AS U
    )
;

 SET (   C_LAST, C_FIRST, C_MIDDLE, 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 )

  = ( SELECT   C_LAST, C_FIRST, C_MIDDLE, 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
           , CASE WHEN C_CREDIT = 'BC' THEN SUBSTR(C_DATA, 1, 200) ELSE NULL END AS
C_DATA

       FROM NEW TABLE ( UPDATE CUSTOMER

                   SET   C_BALANCE    = C_BALANCE - PAY_C_ID.H_AMOUNT
                     , C_YTD_PAYMENT = C_YTD_PAYMENT + PAY_C_ID.H_AMOUNT
                     , C_PAYMENT_CNT = C_PAYMENT_CNT + SMALLINT( 1 )

                     , C_DATA = CASE WHEN C_CREDIT = 'BC'
                                 THEN    BAD_CREDIT_PREFIX       -- 34 bytes long
                                    || SUBSTR( C_DATA, 1, 466 ) -- 466 + 34 = 500 bytes
                                 ELSE C_DATA
                               END

                  WHERE C_W_ID = PAY_C_ID.C_W_ID
                    AND C_D_ID = PAY_C_ID.C_D_ID
                    AND C_ID   = PAY_C_ID.C_ID
                ) AS U
    )
 ;

 SET ( W_NAME, W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP )

  = ( SELECT W_NAME, W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP

      FROM OLD TABLE ( UPDATE WAREHOUSE

                   SET W_YTD = W_YTD + PAY_C_ID.H_AMOUNT

                  WHERE W_ID = PAY_C_ID.W_ID
                ) AS U
   )
 ;

 INSERT

   INTO HISTORY ( H_C_ID, H_C_D_ID, H_C_W_ID, H_D_ID, H_W_ID, H_DATA, H_DATE,
H_AMOUNT )

   VALUES (   PAY_C_ID.C_ID
         , PAY_C_ID.C_D_ID
         , PAY_C_ID.C_W_ID
         , PAY_C_ID.D_ID
         , PAY_C_ID.W_ID
         , VAR.W_NAME || CHAR( '    ', 4 ) || VAR.D_NAME
         , PAY_C_ID.H_DATE
         , PAY_C_ID.H_AMOUNT
        )
 ;

 RETURN VALUES (   W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP
             , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
             , C_LAST, C_FIRST, C_MIDDLE, 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
            )
 ;

END
$

--
-- NEW ORDER
--

CREATE FUNCTION NEW_OL_ALL(  I_ID     INT
                , I_QTY     SMALLINT
                , W_ID      INT
                , SUPP_W_ID INT
                , O_ID      INT
                , D_ID      SMALLINT
                , OL_NUMBER SMALLINT
                )

RETURNS TABLE(   I_PRICE      INTEGER
           , I_NAME       CHAR(24)
           , I_DATA       VARCHAR(50)
           , OL_DIST_INFO  CHAR(24)
           , S_DATA       VARCHAR(50)
           , S_QUANTITY    SMALLINT
          )

SPECIFIC NEW_OL_ALL

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

 DECLARE I_PRICE      INTEGER  ;
 DECLARE I_NAME       CHAR(24)  ;
 DECLARE I_DATA       VARCHAR(50)  ;
 DECLARE OL_DIST_INFO  CHAR(24)  ;
 DECLARE S_DATA       VARCHAR(50)  ;
 DECLARE S_QUANTITY    SMALLINT  ;


 SET   ( I_PRICE , I_NAME , I_DATA )
```

```
  = ( SELECT
            I_PRICE
          , I_NAME
          , I_DATA

      FROM ITEM

     WHERE ITEM.I_ID = NEW_OL_ALL.I_ID
   ) ;

  SET ( OL_DIST_INFO , S_DATA , S_QUANTITY )

   = ( SELECT   OL_DIST_INFO
          , S_DATA
          , S_QUANTITY

       FROM NEW TABLE (    UPDATE STOCK

                    INCLUDE (  OL_DIST_INFO CHAR( 24 ) )

                      SET S_QUANTITY = CASE WHEN S_QUANTITY - NEW_OL_ALL.I_QTY >= 10
                                  THEN S_QUANTITY - NEW_OL_ALL.I_QTY
                                  ELSE S_QUANTITY - NEW_OL_ALL.I_QTY + 91
                             END

                             , S_ORDER_CNT = S_ORDER_CNT + SMALLINT( 1 )

                             , S_YTD = S_YTD + NEW_OL_ALL.I_QTY

                             , S_REMOTE_CNT = CASE WHEN NEW_OL_ALL.SUPP_W_ID =
NEW_OL_ALL.W_ID
                                        THEN S_REMOTE_CNT
                                        ELSE S_REMOTE_CNT + SMALLINT( 1 )
                                   END

                             , OL_DIST_INFO = CASE D_ID WHEN SMALLINT(  1 )  THEN
S_DIST_01
                                          WHEN SMALLINT(  2 )  THEN S_DIST_02
                                          WHEN SMALLINT(  3 )  THEN S_DIST_03
                                          WHEN SMALLINT(  4 )  THEN S_DIST_04
                                          WHEN SMALLINT(  5 )  THEN S_DIST_05
                                          WHEN SMALLINT(  6 )  THEN S_DIST_06
                                          WHEN SMALLINT(  7 )  THEN S_DIST_07
                                          WHEN SMALLINT(  8 )  THEN S_DIST_08
                                          WHEN SMALLINT(  9 )  THEN S_DIST_09
                                          WHEN SMALLINT( 10 )  THEN S_DIST_10
                                    END
                       WHERE S_I_ID = NEW_OL_ALL.I_ID
                         AND S_W_ID = NEW_OL_ALL.SUPP_W_ID
                    ) AS U
     )
   ;

   RETURN VALUES(   VAR.I_PRICE
             , VAR.I_NAME
             , VAR.I_DATA
             , VAR.OL_DIST_INFO
             , VAR.S_DATA
             , VAR.S_QUANTITY
           )
   ;

END
$

CREATE FUNCTION NEW_OL_LOCAL(   I_ID     INT
                      , I_QTY    SMALLINT
                      , W_ID     INT
                      , O_ID     INT
                      , D_ID     SMALLINT
                      , OL_NUMBER SMALLINT
                    )

RETURNS TABLE(   I_PRICE      INTEGER
           , I_NAME       CHAR(24)
           , I_DATA       VARCHAR(50)
           , OL_DIST_INFO  CHAR(24)
           , S_DATA       VARCHAR(50)
           , S_QUANTITY    SMALLINT
         )

SPECIFIC NEW_OL_LOCAL

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

 DECLARE I_PRICE      INTEGER  ;
 DECLARE I_NAME       CHAR(24)  ;
 DECLARE I_DATA       VARCHAR(50)  ;
 DECLARE OL_DIST_INFO  CHAR(24)  ;
 DECLARE S_DATA       VARCHAR(50)  ;
 DECLARE S_QUANTITY    SMALLINT  ;


 SET   ( I_PRICE , I_NAME , I_DATA )

  = ( SELECT
            I_PRICE
          , I_NAME
          , I_DATA

      FROM ITEM

     WHERE ITEM.I_ID = NEW_OL_LOCAL.I_ID
   ) ;

  SET ( OL_DIST_INFO , S_DATA , S_QUANTITY )

   = ( SELECT   OL_DIST_INFO
          , S_DATA
          , S_QUANTITY

       FROM NEW TABLE (    UPDATE STOCK

                    INCLUDE (  OL_DIST_INFO CHAR( 24 ) )

                      SET S_QUANTITY = CASE WHEN S_QUANTITY - NEW_OL_LOCAL.I_QTY
>= 10
                                  THEN S_QUANTITY - NEW_OL_LOCAL.I_QTY
                                  ELSE S_QUANTITY - NEW_OL_LOCAL.I_QTY + 91
                             END

                             , S_ORDER_CNT = S_ORDER_CNT + SMALLINT( 1 )

                             , S_YTD = S_YTD + NEW_OL_LOCAL.I_QTY

                             , OL_DIST_INFO = CASE D_ID WHEN SMALLINT(  1 )  THEN
S_DIST_01
                                          WHEN SMALLINT(  2 )  THEN S_DIST_02
                                          WHEN SMALLINT(  3 )  THEN S_DIST_03
                                          WHEN SMALLINT(  4 )  THEN S_DIST_04
                                          WHEN SMALLINT(  5 )  THEN S_DIST_05
                                          WHEN SMALLINT(  6 )  THEN S_DIST_06
                                          WHEN SMALLINT(  7 )  THEN S_DIST_07
                                          WHEN SMALLINT(  8 )  THEN S_DIST_08
                                          WHEN SMALLINT(  9 )  THEN S_DIST_09
                                          WHEN SMALLINT( 10 )  THEN S_DIST_10
                                    END
                       WHERE S_I_ID = NEW_OL_LOCAL.I_ID
                         AND S_W_ID = NEW_OL_LOCAL.W_ID
                    ) AS U
     )
   ;

   RETURN VALUES(   VAR.I_PRICE
             , VAR.I_NAME
             , VAR.I_DATA
             , VAR.OL_DIST_INFO
             , VAR.S_DATA
             , VAR.S_QUANTITY
           )
   ;

END
$

CREATE FUNCTION NEW_WH (   O_ID       INTEGER
                 , W_ID       INTEGER
                 , D_ID       SMALLINT
                 , C_ID       INTEGER
                 , O_ENTRY_D   BIGINT
                 , O_OL_CNT    SMALLINT
                 , O_ALL_LOCAL SMALLINT
                )

RETURNS TABLE (   W_TAX      INTEGER
           , C_DISCOUNT  INTEGER
           , C_LAST     VARCHAR(16)
           , C_CREDIT    CHAR(2)
         )

SPECIFIC NEW_WH

MODIFIES SQL DATA DETERMINISTIC NO EXTERNAL ACTION LANGUAGE SQL

VAR: BEGIN ATOMIC

 DECLARE C_DISCOUNT  INTEGER ;
 DECLARE C_LAST     VARCHAR(16) ;
 DECLARE C_CREDIT    CHAR(2) ;
 DECLARE W_TAX       INTEGER ;

 INSERT

   INTO NEW_ORDER ( NO_O_ID, NO_D_ID, NO_W_ID )

        VALUES (   O_ID
              , D_ID
              , W_ID
            )
 ;


 INSERT

   INTO ORDERS ( O_C_ID, O_ENTRY_D, O_CARRIER_ID, O_OL_CNT, O_ALL_LOCAL, O_ID,
O_W_ID, O_D_ID )
```

```
      VALUES (   C_ID
               , O_ENTRY_D
               , 0
               , O_OL_CNT
               , O_ALL_LOCAL
               , O_ID
               , W_ID
               , D_ID
              )
 ;

 SET ( C_DISCOUNT, C_LAST, C_CREDIT )

  =  ( SELECT  C_DISCOUNT, C_LAST, C_CREDIT

       FROM  CUSTOMER

      WHERE  C_ID = NEW_WH.C_ID
        AND  C_W_ID = W_ID
        AND  C_D_ID = D_ID
    )
 ;


 SET W_TAX
   =   ( SELECT W_TAX

          FROM WAREHOUSE

         WHERE W_ID = NEW_WH.W_ID
      )
 ;

 RETURN VALUES ( W_TAX , C_DISCOUNT , C_LAST , C_CREDIT ) ;

END

$
```

cat-proc.ddl

```
CREATE PROCEDURE news
                (in new_in   varchar(270) FOR BIT DATA,
                 out new_out varchar(662) FOR BIT DATA)
                LANGUAGE C
                PARAMETER STYLE GENERAL
                EXTERNAL NAME '/home/tpcc/sqllib/function/news!news'
                not fenced;

CREATE PROCEDURE ords
                (in ord_in   varchar(42) FOR BIT DATA,
                 out ord_out varchar(446) FOR BIT DATA)
                LANGUAGE C
                PARAMETER STYLE GENERAL
                EXTERNAL NAME '/home/tpcc/sqllib/function/ords!ords'
                not fenced;

CREATE PROCEDURE dels
                (in del_in   varchar(22) FOR BIT DATA,
                 out del_out varchar(50) FOR BIT DATA)
                LANGUAGE C
                PARAMETER STYLE GENERAL
                EXTERNAL NAME '/home/tpcc/sqllib/function/dels!dels'
                not fenced;
```

dels.exp

```
#! Export file

dels
```

news.exp

```
#! Export file

news
```

ords.exp

```
#! Export file

ords
```

tcc_all_sql.sqc

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * tpcc_all_sql.sqc - Client/Server code for TPCC
 */

#include <stdlib.h>
#include <errno.h>
#include "db2tpcc.h"
#include "tpccapp.h"
#include "tpccdbg.h"

#include "sqlca.h"
#include "sql.h"

#include "lval.h"

// -------------------------------------------------------------
// New Order CLIENT
// -------------------------------------------------------------

static int itemComparison ( const void * a , const void * b )
{
  struct in_items_struct * one = (struct in_items_struct *) a ;
  struct in_items_struct * two = (struct in_items_struct *) b ;

  if ( one->s_OL_I_ID != two->s_OL_I_ID )
  {
    return (  one->s_OL_I_ID - two->s_OL_I_ID  ) ;
  }
  else
  {
    return (  one->s_OL_SUPPLY_W_ID - two->s_OL_SUPPLY_W_ID  ) ;
  }
}

int neword_sql (  struct in_neword_struct  * in_neword
                , struct out_neword_struct * neword      )
{
  struct sqlca sqlca ;

  EXEC SQL BEGIN DECLARE SECTION;

    struct vc_new_in
    {
      short len;
      char data[ 270 ] ;
    } * pHostvarInput ;

    struct vc_new_out
    {
      short len;
      char data[ 662 ] ;
    } * pHostvarOutput ;

  EXEC SQL END DECLARE SECTION;

  int rc = 0;

  int itemIndex = 0 ;
  int actualItemIndex = 0 ;

  in_neword->s_O_ENTRY_D_time = time(NULL) ;
  neword->s_O_ENTRY_D_time = in_neword->s_O_ENTRY_D_time ;

  in_neword->s_all_local  = 1 ;

  for ( itemIndex = 0 ;
        itemIndex < 15 && in_neword->in_item[ itemIndex ].s_OL_I_ID != UNUSED_ITEM_ID ;
        itemIndex++
      )
  {
    if ( in_neword->in_item[ itemIndex ].s_OL_SUPPLY_W_ID != in_neword->s_W_ID )
    {
      in_neword->s_all_local = 0 ;
    }
  }

  in_neword->s_O_OL_CNT = itemIndex ;

  qsort( in_neword->in_item, in_neword->s_O_OL_CNT
       ,  sizeof ( in_neword->in_item[ 0 ] )
```

```
        , itemComparison
      ) ;

  actualItemIndex = -1 ;

  for ( itemIndex = 0
      ; itemIndex < in_neword->s_O_OL_CNT
      ; itemIndex++ )
  {
    if ( actualItemIndex >= 0
        && in_neword->in_item[ actualItemIndex ].s_OL_I_ID       == in_neword->in_item[ itemIndex
].s_OL_I_ID
        && in_neword->in_item[ actualItemIndex ].s_OL_SUPPLY_W_ID == in_neword->in_item[
itemIndex ].s_OL_SUPPLY_W_ID
       )
    {
      in_neword->in_item[ actualItemIndex ].s_OL_QUANTITY  += in_neword->in_item[ itemIndex
].s_OL_QUANTITY ;
    }
    else
    {
      actualItemIndex ++ ;
      in_neword->in_item[ actualItemIndex ].s_OL_I_ID       = in_neword->in_item[ itemIndex
].s_OL_I_ID ;
      in_neword->in_item[ actualItemIndex ].s_OL_SUPPLY_W_ID = in_neword->in_item[ itemIndex
].s_OL_SUPPLY_W_ID ;
      in_neword->in_item[ actualItemIndex ].s_OL_QUANTITY    = in_neword->in_item[ itemIndex
].s_OL_QUANTITY ;

    }
  }

  in_neword->s_O_OL_CNT = actualItemIndex + 1 ;

  pHostvarInput     = (struct vc_new_in *) in_neword ;
  pHostvarInput->len = sizeof(struct in_neword_struct) - SPGENERAL_ADJUST ;

  pHostvarOutput     = (struct vc_new_out *) neword;
  pHostvarOutput->len = sizeof(struct out_neword_struct) - SPGENERAL_ADJUST ;

#ifdef DEBUGIT
  new_debug(neword, in_neword, "Client before SP call");
#endif /* DEBUGIT */

  EXEC SQL CALL news ( :*pHostvarInput, :*pHostvarOutput );

  if ( sqlca.sqlcode == 0 )
  {
    double wtax = neword->s_W_TAX / 10000.0 ;
    double dtax = neword->s_D_TAX / 10000.0 ;
    double cdisc = neword->s_C_DISCOUNT / 10000.0 ;
    double factor = (1.0 - cdisc) * (1.0 + wtax + dtax) ;

    neword->s_total_amount = 0 ;

    for ( itemIndex = 0 ;
        itemIndex < in_neword->s_O_OL_CNT ;
        itemIndex++
      )
    {

      if ( neword->item[ itemIndex ].s_I_PRICE > 0 )
      {
        neword->item[ itemIndex ].s_OL_AMOUNT =  neword->item[ itemIndex ].s_I_PRICE *
                                    in_neword->in_item[ itemIndex ].s_OL_QUANTITY ;

        neword->s_total_amount += neword->item[ itemIndex ].s_OL_AMOUNT ;

      }
    }

    neword->s_total_amount *= factor;

#ifdef DEBUGIT
  new_debug(neword, in_neword, "Client after SP call");
#endif /* DEBUGIT */

  }
  else
  {
    sqlerror( NEWORD_SQL, "NEW", __LINE__, &sqlca) ;
    new_debug( neword, in_neword, "sqlcode != 0" ) ;
    neword->s_transtatus = FATAL_SQLERROR;
    rc = -1 ;
  }

  if (neword->s_transtatus <= FATAL_SQLERROR)
  {
    new_debug(neword, in_neword, "s_transtatus <= FATAL_SQLERROR");
    rc = -1 ;
  }

  if (neword->s_transtatus == INVALID_ITEM) rc = -1;

  return (rc);
}


// ---------------------------------------------------------------------
```

```
// New Order  SERVER
// ---------------------------------------------------------------------

int  static is_ORIGINAL( char *string, short length ) ;


SQL_API_RC new_order_internal( char *pin, char *pout )
{

  struct out_neword_struct *neword;

  struct in_neword_struct  *in_neword;

  struct sqlca sqlca ;

  int     fbadItemDetected = 0 ;

  EXEC SQL BEGIN DECLARE SECTION;

    char    c_last [ 16 ] ;
    char    c_credit [ 2 ] ;
    sqlint32  c_discount ;
    sqlint32  dist_tax ;
    sqlint32  ware_tax ;

    sqlint32   w_id ;
    short      d_id ;
    sqlint32   c_id ;

    sqlint32   next_o_id ;

    short      s_quantity ;

    sqlint32   supply_w_id ;

    short   inputItemCount ;

    char    stockDistrictInformation [ 24 ];
    char    item_name[ 24 ] ;

    sqlint64 o_entry_d ;

    short   allLocal ;

    sqlint32    item_price ;

    struct i_data_type { short len ; char data[ 50 ] ; } i_data ;
    struct s_data_type { short len ; char data[ 50 ] ; } s_data ;

    sqlint32    id0, id1, id2,  id3,  id4,  id5,  id6,  id7;
    sqlint32    id8, id9, id10, id11, id12, id13, id14;

    sqlint32   supply_w_id0,  supply_w_id1, supply_w_id2,  supply_w_id3;
    sqlint32   supply_w_id4,  supply_w_id5, supply_w_id6,  supply_w_id7;
    sqlint32   supply_w_id8,  supply_w_id9, supply_w_id10,  supply_w_id11;
    sqlint32   supply_w_id12,  supply_w_id13, supply_w_id14;

    short   ol_quantity0,  ol_quantity1, ol_quantity2,  ol_quantity3;
    short   ol_quantity4,  ol_quantity5, ol_quantity6,  ol_quantity7;
    short   ol_quantity8,  ol_quantity9, ol_quantity10,  ol_quantity11;
    short   ol_quantity12,  ol_quantity13, ol_quantity14;


  EXEC SQL END DECLARE SECTION;

  int     rc ;
  int     inputItemArrayIndex ;

  char    stockDistrictInformationArray [15][25];

  #define stockDistrictInformation  stockDistrictInformationArray[ inputItemArrayIndex ]

  // Redirected input fields

  #define w_id in_neword->s_W_ID
  #define d_id in_neword->s_D_ID
  #define c_id in_neword->s_C_ID
  #define o_entry_d in_neword->s_O_ENTRY_D_time

  #define inputItemCount in_neword->s_O_OL_CNT

  #define allLocal in_neword->s_all_local

  // Redirected output fields

  #define  c_last       neword->s_C_LAST
  #define  c_credit     neword->s_C_CREDIT
  #define  c_discount   neword->s_C_DISCOUNT
  #define  ware_tax     neword->s_W_TAX
  #define  dist_tax     neword->s_D_TAX
  #define  s_quantity   neword->item[ inputItemArrayIndex ].s_S_QUANTITY

  #define next_o_id neword->s_O_ID


  #define  item_name    neword->item[ inputItemArrayIndex ].s_I_NAME

  sqlint32    i_priceArray[ 15 ] ;
```

```
#define  item_price   i_priceArray[ inputItemArrayIndex ]

struct  i_data_type i_dataArray[ 15 ] ;
struct  s_data_type s_dataArray[ 15 ] ;

#define  i_data       i_dataArray[ inputItemArrayIndex ]
#define  s_data       s_dataArray[ inputItemArrayIndex ]

#define  id0  in_neword->in_item[0].s_OL_I_ID
#define  id1  in_neword->in_item[1].s_OL_I_ID
#define  id2  in_neword->in_item[2].s_OL_I_ID
#define  id3  in_neword->in_item[3].s_OL_I_ID
#define  id4  in_neword->in_item[4].s_OL_I_ID
#define  id5  in_neword->in_item[5].s_OL_I_ID
#define  id6  in_neword->in_item[6].s_OL_I_ID
#define  id7  in_neword->in_item[7].s_OL_I_ID
#define  id8  in_neword->in_item[8].s_OL_I_ID
#define  id9  in_neword->in_item[9].s_OL_I_ID
#define  id10 in_neword->in_item[10].s_OL_I_ID
#define  id11 in_neword->in_item[11].s_OL_I_ID
#define  id12 in_neword->in_item[12].s_OL_I_ID
#define  id13 in_neword->in_item[13].s_OL_I_ID
#define  id14 in_neword->in_item[14].s_OL_I_ID

#define ol_quantity0  in_neword->in_item[  0 ].s_OL_QUANTITY
#define ol_quantity1  in_neword->in_item[  1 ].s_OL_QUANTITY
#define ol_quantity2  in_neword->in_item[  2 ].s_OL_QUANTITY
#define ol_quantity3  in_neword->in_item[  3 ].s_OL_QUANTITY
#define ol_quantity4  in_neword->in_item[  4 ].s_OL_QUANTITY
#define ol_quantity5  in_neword->in_item[  5 ].s_OL_QUANTITY
#define ol_quantity6  in_neword->in_item[  6 ].s_OL_QUANTITY
#define ol_quantity7  in_neword->in_item[  7 ].s_OL_QUANTITY
#define ol_quantity8  in_neword->in_item[  8 ].s_OL_QUANTITY
#define ol_quantity9  in_neword->in_item[  9 ].s_OL_QUANTITY
#define ol_quantity10 in_neword->in_item[ 10 ].s_OL_QUANTITY
#define ol_quantity11 in_neword->in_item[ 11 ].s_OL_QUANTITY
#define ol_quantity12 in_neword->in_item[ 12 ].s_OL_QUANTITY
#define ol_quantity13 in_neword->in_item[ 13 ].s_OL_QUANTITY
#define ol_quantity14 in_neword->in_item[ 14 ].s_OL_QUANTITY

#define supply_w_id0  in_neword->in_item[  0 ].s_OL_SUPPLY_W_ID
#define supply_w_id1  in_neword->in_item[  1 ].s_OL_SUPPLY_W_ID
#define supply_w_id2  in_neword->in_item[  2 ].s_OL_SUPPLY_W_ID
#define supply_w_id3  in_neword->in_item[  3 ].s_OL_SUPPLY_W_ID
#define supply_w_id4  in_neword->in_item[  4 ].s_OL_SUPPLY_W_ID
#define supply_w_id5  in_neword->in_item[  5 ].s_OL_SUPPLY_W_ID
#define supply_w_id6  in_neword->in_item[  6 ].s_OL_SUPPLY_W_ID
#define supply_w_id7  in_neword->in_item[  7 ].s_OL_SUPPLY_W_ID
#define supply_w_id8  in_neword->in_item[  8 ].s_OL_SUPPLY_W_ID
#define supply_w_id9  in_neword->in_item[  9 ].s_OL_SUPPLY_W_ID
#define supply_w_id10 in_neword->in_item[ 10 ].s_OL_SUPPLY_W_ID
#define supply_w_id11 in_neword->in_item[ 11 ].s_OL_SUPPLY_W_ID
#define supply_w_id12 in_neword->in_item[ 12 ].s_OL_SUPPLY_W_ID
#define supply_w_id13 in_neword->in_item[ 13 ].s_OL_SUPPLY_W_ID
#define supply_w_id14 in_neword->in_item[ 14 ].s_OL_SUPPLY_W_ID

EXEC SQL DECLARE ISOL_Remote_1 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

                (   OL_O_ID
                  , OL_D_ID
                  , OL_W_ID
                  , OL_NUMBER
                  , OL_I_ID
                  , OL_SUPPLY_W_ID
                  , OL_DELIVERY_D
                  , OL_QUANTITY
                  , OL_AMOUNT
                  , OL_DIST_INFO
                )

                INCLUDE (   I_PRICE    INTEGER
                          , I_NAME     CHAR(24)
                          , I_DATA     VARCHAR( 50 )
                          , S_DATA     VARCHAR( 50 )
                          , S_QUANTITY SMALLINT
                        )

                     SELECT   O_ID
                           , D_ID
                           , W_ID
                           , OL_NUMBER
                           , I_ID
                           , I_SUPPLY_W_ID
                           , 0 AS OL_DELIVERY_D
                           , I_QTY
                           , ( I_PRICE * I_QTY )
                           , OL_DIST_INFO
                           , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                      FROM ( SELECT   :next_o_id as O_ID
                                    , :d_id as D_ID
                                    , :w_id AS W_ID
                                    , OL_NUMBER
                                    , I_ID
                                    , I_SUPPLY_W_ID
                                    , I_QTY

                               FROM  TABLE( VALUES

                                          (  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )

                                        ) AS X ( OL_NUMBER , I_ID  , I_QTY         , I_SUPPLY_W_ID  )
                             ) AS ITEMLIST

                           , TABLE ( NEW_OL_ALL(   I_ID
                                                 , I_QTY
                                                 , W_ID
                                                 , I_SUPPLY_W_ID
                                                 , O_ID
                                                 , D_ID
                                                 , OL_NUMBER
                                                )
                                   ) AS NEW_OL_ALL

                      WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

                ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_2 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

                (   OL_O_ID
                  , OL_D_ID
                  , OL_W_ID
                  , OL_NUMBER
                  , OL_I_ID
                  , OL_SUPPLY_W_ID
                  , OL_DELIVERY_D
                  , OL_QUANTITY
                  , OL_AMOUNT
                  , OL_DIST_INFO
                )

                INCLUDE (   I_PRICE    INTEGER
                          , I_NAME     CHAR(24)
                          , I_DATA     VARCHAR( 50 )
                          , S_DATA     VARCHAR( 50 )
                          , S_QUANTITY SMALLINT
                        )

                     SELECT   O_ID
                           , D_ID
                           , W_ID
                           , OL_NUMBER
                           , I_ID
                           , I_SUPPLY_W_ID
                           , 0 AS OL_DELIVERY_D
                           , I_QTY
                           , ( I_PRICE * I_QTY )
                           , OL_DIST_INFO
                           , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                      FROM ( SELECT   :next_o_id as O_ID
                                    , :d_id as D_ID
                                    , :w_id AS W_ID
                                    , OL_NUMBER
                                    , I_ID
                                    , I_SUPPLY_W_ID
                                    , I_QTY

                               FROM  TABLE( VALUES

                                          (  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                        , (  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )

                                        ) AS X ( OL_NUMBER , I_ID  , I_QTY         , I_SUPPLY_W_ID  )
                             ) AS ITEMLIST

                           , TABLE ( NEW_OL_ALL(   I_ID
                                                 , I_QTY
                                                 , W_ID
                                                 , I_SUPPLY_W_ID
                                                 , O_ID
                                                 , D_ID
                                                 , OL_NUMBER
                                                )
                                   ) AS NEW_OL_ALL

                      WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

                ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_3  CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

                (   OL_O_ID
```

```
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR( 50 )
                        , S_DATA     VARCHAR( 50 )
                        , S_QUANTITY SMALLINT
                      )

                   SELECT   O_ID
                        , D_ID
                        , W_ID
                        , OL_NUMBER
                        , I_ID
                        , I_SUPPLY_W_ID
                        , 0 AS OL_DELIVERY_D
                        , I_QTY
                        , ( I_PRICE * I_QTY )
                        , OL_DIST_INFO
                        , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                    FROM ( SELECT   :next_o_id as O_ID
                                , :d_id as D_ID
                                , :w_id AS W_ID
                                , OL_NUMBER
                                , I_ID
                                , I_SUPPLY_W_ID
                                , I_QTY

                             FROM  TABLE( VALUES

                                        (  SMALLINT( 1 )        , :id0  , :ol_quantity0  , :supply_w_id0  )
                                      , (  SMALLINT( 2 )        , :id1  , :ol_quantity1  , :supply_w_id1  )
                                      , (  SMALLINT( 3 )        , :id2  , :ol_quantity2  , :supply_w_id2  )

                                    ) AS X ( OL_NUMBER , I_ID  , I_QTY         , I_SUPPLY_W_ID  )
                           ) AS ITEMLIST

                         , TABLE ( NEW_OL_ALL(   I_ID
                                               , I_QTY
                                               , W_ID
                                               , I_SUPPLY_W_ID
                                               , O_ID
                                               , D_ID
                                               , OL_NUMBER
                                             )
                                 ) AS NEW_OL_ALL

                    WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_4  CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR( 50 )
                        , S_DATA     VARCHAR( 50 )
                        , S_QUANTITY SMALLINT
                      )

                   SELECT   O_ID
                        , D_ID
                        , W_ID
                        , OL_NUMBER
                        , I_ID
                        , I_SUPPLY_W_ID
                        , 0 AS OL_DELIVERY_D
                        , I_QTY
                        , ( I_PRICE * I_QTY )
                        , OL_DIST_INFO
                        , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY
```

```
                    FROM ( SELECT   :next_o_id as O_ID
                                , :d_id as D_ID
                                , :w_id AS W_ID
                                , OL_NUMBER
                                , I_ID
                                , I_SUPPLY_W_ID
                                , I_QTY

                             FROM  TABLE( VALUES

                                        (  SMALLINT( 1 )        , :id0  , :ol_quantity0  , :supply_w_id0  )
                                      , (  SMALLINT( 2 )        , :id1  , :ol_quantity1  , :supply_w_id1  )
                                      , (  SMALLINT( 3 )        , :id2  , :ol_quantity2  , :supply_w_id2  )
                                      , (  SMALLINT( 4 )        , :id3  , :ol_quantity3  , :supply_w_id3  )

                                    ) AS X ( OL_NUMBER , I_ID  , I_QTY         , I_SUPPLY_W_ID  )
                           ) AS ITEMLIST

                         , TABLE ( NEW_OL_ALL(   I_ID
                                               , I_QTY
                                               , W_ID
                                               , I_SUPPLY_W_ID
                                               , O_ID
                                               , D_ID
                                               , OL_NUMBER
                                             )
                                 ) AS NEW_OL_ALL

                    WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_5  CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR( 50 )
                        , S_DATA     VARCHAR( 50 )
                        , S_QUANTITY SMALLINT
                      )

                   SELECT   O_ID
                        , D_ID
                        , W_ID
                        , OL_NUMBER
                        , I_ID
                        , I_SUPPLY_W_ID
                        , 0 AS OL_DELIVERY_D
                        , I_QTY
                        , ( I_PRICE * I_QTY )
                        , OL_DIST_INFO
                        , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                    FROM ( SELECT   :next_o_id as O_ID
                                , :d_id as D_ID
                                , :w_id AS W_ID
                                , OL_NUMBER
                                , I_ID
                                , I_SUPPLY_W_ID
                                , I_QTY

                             FROM  TABLE( VALUES

                                        (  SMALLINT( 1 )        , :id0  , :ol_quantity0  , :supply_w_id0  )
                                      , (  SMALLINT( 2 )        , :id1  , :ol_quantity1  , :supply_w_id1  )
                                      , (  SMALLINT( 3 )        , :id2  , :ol_quantity2  , :supply_w_id2  )
                                      , (  SMALLINT( 4 )        , :id3  , :ol_quantity3  , :supply_w_id3  )
                                      , (  SMALLINT( 5 )        , :id4  , :ol_quantity4  , :supply_w_id4  )

                                    ) AS X ( OL_NUMBER , I_ID  , I_QTY         , I_SUPPLY_W_ID  )
                           ) AS ITEMLIST

                         , TABLE ( NEW_OL_ALL(   I_ID
                                               , I_QTY
                                               , W_ID
                                               , I_SUPPLY_W_ID
                                               , O_ID
                                               , D_ID
                                               , OL_NUMBER
                                             )
                                 ) AS NEW_OL_ALL
```

```
                    WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_6 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR( 50 )
                        , S_DATA     VARCHAR( 50 )
                        , S_QUANTITY SMALLINT
                      )

                   SELECT   O_ID
                        , D_ID
                        , W_ID
                        , OL_NUMBER
                        , I_ID
                        , I_SUPPLY_W_ID
                        , 0 AS OL_DELIVERY_D
                        , I_QTY
                        , ( I_PRICE * I_QTY )
                        , OL_DIST_INFO
                        , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                   FROM ( SELECT   :next_o_id as O_ID
                                 , :d_id as D_ID
                                 , :w_id AS W_ID
                                 , OL_NUMBER
                                 , I_ID
                                 , I_SUPPLY_W_ID
                                 , I_QTY

                              FROM  TABLE( VALUES

                                        (  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                      , (  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                                      , (  SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                                      , (  SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                                      , (  SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                                      , (  SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )

                                    ) AS X ( OL_NUMBER , I_ID  , I_QTY         , I_SUPPLY_W_ID  )
                          ) AS ITEMLIST

                        , TABLE ( NEW_OL_ALL(   I_ID
                                              , I_QTY
                                              , W_ID
                                              , I_SUPPLY_W_ID
                                              , O_ID
                                              , D_ID
                                              , OL_NUMBER
                                            )
                                ) AS NEW_OL_ALL

                   WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_7 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR( 50 )
                        , S_DATA     VARCHAR( 50 )
                        , S_QUANTITY SMALLINT
                      )

                   SELECT   O_ID
                        , D_ID
                        , W_ID
                        , OL_NUMBER
                        , I_ID
                        , I_SUPPLY_W_ID
                        , 0 AS OL_DELIVERY_D
                        , I_QTY
                        , ( I_PRICE * I_QTY )
                        , OL_DIST_INFO
                        , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                   FROM ( SELECT   :next_o_id as O_ID
                                 , :d_id as D_ID
                                 , :w_id AS W_ID
                                 , OL_NUMBER
                                 , I_ID
                                 , I_SUPPLY_W_ID
                                 , I_QTY

                              FROM  TABLE( VALUES

                                        (  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                      , (  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                                      , (  SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                                      , (  SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                                      , (  SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                                      , (  SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )

                                    ) AS X ( OL_NUMBER , I_ID  , I_QTY         , I_SUPPLY_W_ID  )
                          ) AS ITEMLIST

                        , TABLE ( NEW_OL_ALL(   I_ID
                                              , I_QTY
                                              , W_ID
                                              , I_SUPPLY_W_ID
                                              , O_ID
                                              , D_ID
                                              , OL_NUMBER
                                            )
                                ) AS NEW_OL_ALL

                   WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_8 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR( 50 )
                        , S_DATA     VARCHAR( 50 )
                        , S_QUANTITY SMALLINT
                      )

                   SELECT   O_ID
                        , D_ID
                        , W_ID
                        , OL_NUMBER
                        , I_ID
                        , I_SUPPLY_W_ID
                        , 0 AS OL_DELIVERY_D
                        , I_QTY
                        , ( I_PRICE * I_QTY )
                        , OL_DIST_INFO
                        , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                   FROM ( SELECT   :next_o_id as O_ID
                                 , :d_id as D_ID
                                 , :w_id AS W_ID
                                 , OL_NUMBER
                                 , I_ID
                                 , I_SUPPLY_W_ID
                                 , I_QTY

                              FROM  TABLE( VALUES

                                        (  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
```

```
                                , (  SMALLINT( 2 )        , :id1  , :ol_quantity1  , :supply_w_id1  )
                                , (  SMALLINT( 3 )        , :id2  , :ol_quantity2  , :supply_w_id2  )
                                , (  SMALLINT( 4 )        , :id3  , :ol_quantity3  , :supply_w_id3  )
                                , (  SMALLINT( 5 )        , :id4  , :ol_quantity4  , :supply_w_id4  )
                                , (  SMALLINT( 6 )        , :id5  , :ol_quantity5  , :supply_w_id5  )
                                , (  SMALLINT( 7 )        , :id6  , :ol_quantity6  , :supply_w_id6  )
                                , (  SMALLINT( 8 )        , :id7  , :ol_quantity7  , :supply_w_id7  )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY         , I_SUPPLY_W_ID  )
                            ) AS ITEMLIST

                            , TABLE ( NEW_OL_ALL(  I_ID
                                        , I_QTY
                                        , W_ID
                                        , I_SUPPLY_W_ID
                                        , O_ID
                                        , D_ID
                                        , OL_NUMBER
                                       )
                                    ) AS NEW_OL_ALL

                            WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

                        ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_9  CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

                ( OL_O_ID
                  , OL_D_ID
                  , OL_W_ID
                  , OL_NUMBER
                  , OL_I_ID
                  , OL_SUPPLY_W_ID
                  , OL_DELIVERY_D
                  , OL_QUANTITY
                  , OL_AMOUNT
                  , OL_DIST_INFO
                )

                INCLUDE (  I_PRICE   INTEGER
                        , I_NAME    CHAR(24)
                        , I_DATA    VARCHAR( 50 )
                        , S_DATA    VARCHAR( 50 )
                        , S_QUANTITY SMALLINT
                       )

                    SELECT  O_ID
                         , D_ID
                         , W_ID
                         , OL_NUMBER
                         , I_ID
                         , I_SUPPLY_W_ID
                         , 0 AS OL_DELIVERY_D
                         , I_QTY
                         , ( I_PRICE * I_QTY )
                         , OL_DIST_INFO
                         , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                    FROM ( SELECT  :next_o_id as O_ID
                                , :d_id as D_ID
                                , :w_id AS W_ID
                                , OL_NUMBER
                                , I_ID
                                , I_SUPPLY_W_ID
                                , I_QTY

                           FROM  TABLE( VALUES

                                  (  SMALLINT( 1 )        , :id0  , :ol_quantity0  , :supply_w_id0  )
                                , (  SMALLINT( 2 )        , :id1  , :ol_quantity1  , :supply_w_id1  )
                                , (  SMALLINT( 3 )        , :id2  , :ol_quantity2  , :supply_w_id2  )
                                , (  SMALLINT( 4 )        , :id3  , :ol_quantity3  , :supply_w_id3  )
                                , (  SMALLINT( 5 )        , :id4  , :ol_quantity4  , :supply_w_id4  )
                                , (  SMALLINT( 6 )        , :id5  , :ol_quantity5  , :supply_w_id5  )
                                , (  SMALLINT( 7 )        , :id6  , :ol_quantity6  , :supply_w_id6  )
                                , (  SMALLINT( 8 )        , :id7  , :ol_quantity7  , :supply_w_id7  )
                                , (  SMALLINT( 9 )        , :id8  , :ol_quantity8  , :supply_w_id8  )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY         , I_SUPPLY_W_ID  )
                            ) AS ITEMLIST

                            , TABLE ( NEW_OL_ALL(  I_ID
                                        , I_QTY
                                        , W_ID
                                        , I_SUPPLY_W_ID
                                        , O_ID
                                        , D_ID
                                        , OL_NUMBER
                                       )
                                    ) AS NEW_OL_ALL

                            WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

                        ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_10 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

                ( OL_O_ID
                  , OL_D_ID
                  , OL_W_ID
                  , OL_NUMBER
                  , OL_I_ID
                  , OL_SUPPLY_W_ID
                  , OL_DELIVERY_D
                  , OL_QUANTITY
                  , OL_AMOUNT
                  , OL_DIST_INFO
                )

                INCLUDE (  I_PRICE   INTEGER
                        , I_NAME    CHAR(24)
                        , I_DATA    VARCHAR( 50 )
                        , S_DATA    VARCHAR( 50 )
                        , S_QUANTITY SMALLINT
                       )

                    SELECT  O_ID
                         , D_ID
                         , W_ID
                         , OL_NUMBER
                         , I_ID
                         , I_SUPPLY_W_ID
                         , 0 AS OL_DELIVERY_D
                         , I_QTY
                         , ( I_PRICE * I_QTY )
                         , OL_DIST_INFO
                         , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                    FROM ( SELECT  :next_o_id as O_ID
                                , :d_id as D_ID
                                , :w_id AS W_ID
                                , OL_NUMBER
                                , I_ID
                                , I_SUPPLY_W_ID
                                , I_QTY

                           FROM  TABLE( VALUES

                                  (  SMALLINT( 1 )        , :id0  , :ol_quantity0  , :supply_w_id0  )
                                , (  SMALLINT( 2 )        , :id1  , :ol_quantity1  , :supply_w_id1  )
                                , (  SMALLINT( 3 )        , :id2  , :ol_quantity2  , :supply_w_id2  )
                                , (  SMALLINT( 4 )        , :id3  , :ol_quantity3  , :supply_w_id3  )
                                , (  SMALLINT( 5 )        , :id4  , :ol_quantity4  , :supply_w_id4  )
                                , (  SMALLINT( 6 )        , :id5  , :ol_quantity5  , :supply_w_id5  )
                                , (  SMALLINT( 7 )        , :id6  , :ol_quantity6  , :supply_w_id6  )
                                , (  SMALLINT( 8 )        , :id7  , :ol_quantity7  , :supply_w_id7  )
                                , (  SMALLINT( 9 )        , :id8  , :ol_quantity8  , :supply_w_id8  )
                                , (  SMALLINT( 10 )       , :id9  , :ol_quantity9  , :supply_w_id9  )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY         , I_SUPPLY_W_ID  )
                            ) AS ITEMLIST

                            , TABLE ( NEW_OL_ALL(  I_ID
                                        , I_QTY
                                        , W_ID
                                        , I_SUPPLY_W_ID
                                        , O_ID
                                        , D_ID
                                        , OL_NUMBER
                                       )
                                    ) AS NEW_OL_ALL

                            WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

                        ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_11 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

                ( OL_O_ID
                  , OL_D_ID
                  , OL_W_ID
                  , OL_NUMBER
                  , OL_I_ID
                  , OL_SUPPLY_W_ID
                  , OL_DELIVERY_D
                  , OL_QUANTITY
                  , OL_AMOUNT
                  , OL_DIST_INFO
                )

                INCLUDE (  I_PRICE   INTEGER
                        , I_NAME    CHAR(24)
                        , I_DATA    VARCHAR( 50 )
```

```
                    , S_DATA     VARCHAR( 50 )
                    , S_QUANTITY SMALLINT
                   )

                   SELECT   O_ID
                        , D_ID
                        , W_ID
                        , OL_NUMBER
                        , I_ID
                        , I_SUPPLY_W_ID
                        , 0 AS OL_DELIVERY_D
                        , I_QTY
                        , ( I_PRICE * I_QTY )
                        , OL_DIST_INFO
                        , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                   FROM ( SELECT   :next_o_id as O_ID
                             , :d_id as D_ID
                             , :w_id AS W_ID
                             , OL_NUMBER
                             , I_ID
                             , I_SUPPLY_W_ID
                             , I_QTY

                          FROM  TABLE( VALUES

                                 (  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                , (  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                                , (  SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                                , (  SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                                , (  SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                                , (  SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
                                , (  SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )
                                , (  SMALLINT( 8 )       , :id7  , :ol_quantity7  , :supply_w_id7  )
                                , (  SMALLINT( 9 )       , :id8  , :ol_quantity8  , :supply_w_id8  )
                                , (  SMALLINT( 10 )      , :id9  , :ol_quantity9  , :supply_w_id9  )
                                , (  SMALLINT( 11 )      , :id10 , :ol_quantity10 , :supply_w_id10 )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY        , I_SUPPLY_W_ID )
                          ) AS ITEMLIST

                        , TABLE ( NEW_OL_ALL(  I_ID
                                     , I_QTY
                                     , W_ID
                                     , I_SUPPLY_W_ID
                                     , O_ID
                                     , D_ID
                                     , OL_NUMBER
                                    )
                                ) AS NEW_OL_ALL

                   WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

              ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_12 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (  OL_O_ID
               , OL_D_ID
               , OL_W_ID
               , OL_NUMBER
               , OL_I_ID
               , OL_SUPPLY_W_ID
               , OL_DELIVERY_D
               , OL_QUANTITY
               , OL_AMOUNT
               , OL_DIST_INFO
              )

              INCLUDE (  I_PRICE    INTEGER
                    , I_NAME     CHAR(24)
                    , I_DATA     VARCHAR( 50 )
                    , S_DATA     VARCHAR( 50 )
                    , S_QUANTITY SMALLINT
                   )

                   SELECT   O_ID
                        , D_ID
                        , W_ID
                        , OL_NUMBER
                        , I_ID
                        , I_SUPPLY_W_ID
                        , 0 AS OL_DELIVERY_D
                        , I_QTY
                        , ( I_PRICE * I_QTY )
                        , OL_DIST_INFO
                        , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                   FROM ( SELECT   :next_o_id as O_ID
                             , :d_id as D_ID
                             , :w_id AS W_ID
                             , OL_NUMBER
                             , I_ID
                             , I_SUPPLY_W_ID
                             , I_QTY

                          FROM  TABLE( VALUES

                                 (  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                , (  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                                , (  SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                                , (  SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                                , (  SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                                , (  SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
                                , (  SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )
                                , (  SMALLINT( 8 )       , :id7  , :ol_quantity7  , :supply_w_id7  )
                                , (  SMALLINT( 9 )       , :id8  , :ol_quantity8  , :supply_w_id8  )
                                , (  SMALLINT( 10 )      , :id9  , :ol_quantity9  , :supply_w_id9  )
                                , (  SMALLINT( 11 )      , :id10 , :ol_quantity10 , :supply_w_id10 )
                                , (  SMALLINT( 12 )      , :id11 , :ol_quantity11 , :supply_w_id11 )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY        , I_SUPPLY_W_ID )
                          ) AS ITEMLIST

                        , TABLE ( NEW_OL_ALL(  I_ID
                                     , I_QTY
                                     , W_ID
                                     , I_SUPPLY_W_ID
                                     , O_ID
                                     , D_ID
                                     , OL_NUMBER
                                    )
                                ) AS NEW_OL_ALL

                   WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

              ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_13 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (  OL_O_ID
               , OL_D_ID
               , OL_W_ID
               , OL_NUMBER
               , OL_I_ID
               , OL_SUPPLY_W_ID
               , OL_DELIVERY_D
               , OL_QUANTITY
               , OL_AMOUNT
               , OL_DIST_INFO
              )

              INCLUDE (  I_PRICE    INTEGER
                    , I_NAME     CHAR(24)
                    , I_DATA     VARCHAR( 50 )
                    , S_DATA     VARCHAR( 50 )
                    , S_QUANTITY SMALLINT
                   )

                   SELECT   O_ID
                        , D_ID
                        , W_ID
                        , OL_NUMBER
                        , I_ID
                        , I_SUPPLY_W_ID
                        , 0 AS OL_DELIVERY_D
                        , I_QTY
                        , ( I_PRICE * I_QTY )
                        , OL_DIST_INFO
                        , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                   FROM ( SELECT   :next_o_id as O_ID
                             , :d_id as D_ID
                             , :w_id AS W_ID
                             , OL_NUMBER
                             , I_ID
                             , I_SUPPLY_W_ID
                             , I_QTY

                          FROM  TABLE( VALUES

                                 (  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                                , (  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                                , (  SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                                , (  SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                                , (  SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                                , (  SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
                                , (  SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )
                                , (  SMALLINT( 8 )       , :id7  , :ol_quantity7  , :supply_w_id7  )
                                , (  SMALLINT( 9 )       , :id8  , :ol_quantity8  , :supply_w_id8  )
                                , (  SMALLINT( 10 )      , :id9  , :ol_quantity9  , :supply_w_id9  )
                                , (  SMALLINT( 11 )      , :id10 , :ol_quantity10 , :supply_w_id10 )
                                , (  SMALLINT( 12 )      , :id11 , :ol_quantity11 , :supply_w_id11 )
                                , (  SMALLINT( 13 )      , :id12 , :ol_quantity12 , :supply_w_id12 )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY        , I_SUPPLY_W_ID )
                          ) AS ITEMLIST
```

```
                    , TABLE ( NEW_OL_ALL(   I_ID
                                          , I_QTY
                                          , W_ID
                                          , I_SUPPLY_W_ID
                                          , O_ID
                                          , D_ID
                                          , OL_NUMBER
                                          )
                            ) AS NEW_OL_ALL

                WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

        ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_14 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

            (   OL_O_ID
              , OL_D_ID
              , OL_W_ID
              , OL_NUMBER
              , OL_I_ID
              , OL_SUPPLY_W_ID
              , OL_DELIVERY_D
              , OL_QUANTITY
              , OL_AMOUNT
              , OL_DIST_INFO
            )

            INCLUDE (   I_PRICE    INTEGER
                      , I_NAME     CHAR(24)
                      , I_DATA     VARCHAR( 50 )
                      , S_DATA     VARCHAR( 50 )
                      , S_QUANTITY SMALLINT
                    )

                SELECT   O_ID
                       , D_ID
                       , W_ID
                       , OL_NUMBER
                       , I_ID
                       , I_SUPPLY_W_ID
                       , 0 AS OL_DELIVERY_D
                       , I_QTY
                       , ( I_PRICE * I_QTY )
                       , OL_DIST_INFO
                       , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                FROM ( SELECT   :next_o_id as O_ID
                              , :d_id as D_ID
                              , :w_id AS W_ID
                              , OL_NUMBER
                              , I_ID
                              , I_SUPPLY_W_ID
                              , I_QTY

                         FROM  TABLE( VALUES

                                 (  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                               , (  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                               , (  SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                               , (  SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                               , (  SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                               , (  SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
                               , (  SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )
                               , (  SMALLINT( 8 )       , :id7  , :ol_quantity7  , :supply_w_id7  )
                               , (  SMALLINT( 9 )       , :id8  , :ol_quantity8  , :supply_w_id8  )
                               , (  SMALLINT( 10 )      , :id9  , :ol_quantity9  , :supply_w_id9  )
                               , (  SMALLINT( 11 )      , :id10 , :ol_quantity10 , :supply_w_id10 )
                               , (  SMALLINT( 12 )      , :id11 , :ol_quantity11 , :supply_w_id11 )
                               , (  SMALLINT( 13 )      , :id12 , :ol_quantity12 , :supply_w_id12 )
                               , (  SMALLINT( 14 )      , :id13 , :ol_quantity13 , :supply_w_id13 )

                             ) AS X ( OL_NUMBER , I_ID  , I_QTY         , I_SUPPLY_W_ID )
                    ) AS ITEMLIST

                  , TABLE ( NEW_OL_ALL(   I_ID
                                        , I_QTY
                                        , W_ID
                                        , I_SUPPLY_W_ID
                                        , O_ID
                                        , D_ID
                                        , OL_NUMBER
                                        )
                          ) AS NEW_OL_ALL

                WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

        ) AS INS
;

EXEC SQL DECLARE ISOL_Remote_15 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

            (   OL_O_ID
              , OL_D_ID
              , OL_W_ID
              , OL_NUMBER
              , OL_I_ID
              , OL_SUPPLY_W_ID
              , OL_DELIVERY_D
              , OL_QUANTITY
              , OL_AMOUNT
              , OL_DIST_INFO
            )

            INCLUDE (   I_PRICE    INTEGER
                      , I_NAME     CHAR(24)
                      , I_DATA     VARCHAR( 50 )
                      , S_DATA     VARCHAR( 50 )
                      , S_QUANTITY SMALLINT
                    )

                SELECT   O_ID
                       , D_ID
                       , W_ID
                       , OL_NUMBER
                       , I_ID
                       , I_SUPPLY_W_ID
                       , 0 AS OL_DELIVERY_D
                       , I_QTY
                       , ( I_PRICE * I_QTY )
                       , OL_DIST_INFO
                       , I_PRICE , I_NAME , I_DATA , S_DATA , S_QUANTITY

                FROM ( SELECT   :next_o_id as O_ID
                              , :d_id as D_ID
                              , :w_id AS W_ID
                              , OL_NUMBER
                              , I_ID
                              , I_SUPPLY_W_ID
                              , I_QTY

                         FROM  TABLE( VALUES

                                 (  SMALLINT( 1 )       , :id0  , :ol_quantity0  , :supply_w_id0  )
                               , (  SMALLINT( 2 )       , :id1  , :ol_quantity1  , :supply_w_id1  )
                               , (  SMALLINT( 3 )       , :id2  , :ol_quantity2  , :supply_w_id2  )
                               , (  SMALLINT( 4 )       , :id3  , :ol_quantity3  , :supply_w_id3  )
                               , (  SMALLINT( 5 )       , :id4  , :ol_quantity4  , :supply_w_id4  )
                               , (  SMALLINT( 6 )       , :id5  , :ol_quantity5  , :supply_w_id5  )
                               , (  SMALLINT( 7 )       , :id6  , :ol_quantity6  , :supply_w_id6  )
                               , (  SMALLINT( 8 )       , :id7  , :ol_quantity7  , :supply_w_id7  )
                               , (  SMALLINT( 9 )       , :id8  , :ol_quantity8  , :supply_w_id8  )
                               , (  SMALLINT( 10 )      , :id9  , :ol_quantity9  , :supply_w_id9  )
                               , (  SMALLINT( 11 )      , :id10 , :ol_quantity10 , :supply_w_id10 )
                               , (  SMALLINT( 12 )      , :id11 , :ol_quantity11 , :supply_w_id11 )
                               , (  SMALLINT( 13 )      , :id12 , :ol_quantity12 , :supply_w_id12 )
                               , (  SMALLINT( 14 )      , :id13 , :ol_quantity13 , :supply_w_id13 )
                               , (  SMALLINT( 15 )      , :id14 , :ol_quantity14 , :supply_w_id14 )

                             ) AS X ( OL_NUMBER , I_ID  , I_QTY         , I_SUPPLY_W_ID )
                    ) AS ITEMLIST

                  , TABLE ( NEW_OL_ALL(   I_ID
                                        , I_QTY
                                        , W_ID
                                        , I_SUPPLY_W_ID
                                        , O_ID
                                        , D_ID
                                        , OL_NUMBER
                                        )
                          ) AS NEW_OL_ALL

                WHERE NEW_OL_ALL.I_PRICE IS NOT NULL

        ) AS INS
;

EXEC SQL DECLARE ISOL_Local_1  CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

            (   OL_O_ID
              , OL_D_ID
              , OL_W_ID
              , OL_NUMBER
              , OL_I_ID
              , OL_SUPPLY_W_ID
              , OL_DELIVERY_D
              , OL_QUANTITY
              , OL_AMOUNT
              , OL_DIST_INFO
            )

            INCLUDE (   I_PRICE    INTEGER
                      , I_NAME     CHAR(24)
                      , I_DATA     VARCHAR(50)
                      , S_DATA     VARCHAR(50)
```

```
                    , S_QUANTITY SMALLINT )


                  SELECT   O_ID
                       , D_ID
                       , W_ID
                       , OL_NUMBER
                       , I_ID
                       , W_ID AS I_SUPPLY_W_ID
                       , 0 AS OL_DELIVERY_D
                       , I_QTY
                       , ( I_PRICE * I_QTY )
                       , OL_DIST_INFO
                       , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                   FROM ( SELECT   :next_o_id as O_ID
                              , :w_id AS W_ID
                              , :d_id as D_ID
                              , OL_NUMBER
                              , I_ID
                              , I_QTY

                          FROM  Table( VALUES

                                  ( SMALLINT( 1 )   , :id0  , :ol_quantity0  )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                        ) AS ITEMLIST

                    , TABLE( NEW_OL_LOCAL(   I_ID
                                  , I_QTY
                                  , W_ID
                                  , O_ID
                                  , D_ID
                                  , OL_NUMBER
                                  )
                          ) AS NEW_OL_LOCAL

                  WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Local_2  CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                    , I_NAME     CHAR(24)
                    , I_DATA     VARCHAR(50)
                    , S_DATA     VARCHAR(50)
                    , S_QUANTITY SMALLINT )


                  SELECT   O_ID
                       , D_ID
                       , W_ID
                       , OL_NUMBER
                       , I_ID
                       , W_ID AS I_SUPPLY_W_ID
                       , 0 AS OL_DELIVERY_D
                       , I_QTY
                       , ( I_PRICE * I_QTY )
                       , OL_DIST_INFO
                       , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                   FROM ( SELECT   :next_o_id as O_ID
                              , :w_id AS W_ID
                              , :d_id as D_ID
                              , OL_NUMBER
                              , I_ID
                              , I_QTY

                          FROM  Table( VALUES

                                  ( SMALLINT( 1 ) , :id0  , :ol_quantity0  )
                                , ( SMALLINT( 2 ) , :id1  , :ol_quantity1  )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                        ) AS ITEMLIST

                    , TABLE( NEW_OL_LOCAL(   I_ID
                                  , I_QTY
                                  , W_ID
                                  , O_ID
                                  , D_ID
                                  , OL_NUMBER
                                  )
                          ) AS NEW_OL_LOCAL

                  WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Local_3  CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                    , I_NAME     CHAR(24)
                    , I_DATA     VARCHAR(50)
                    , S_DATA     VARCHAR(50)
                    , S_QUANTITY SMALLINT )


                  SELECT   O_ID
                       , D_ID
                       , W_ID
                       , OL_NUMBER
                       , I_ID
                       , W_ID AS I_SUPPLY_W_ID
                       , 0 AS OL_DELIVERY_D
                       , I_QTY
                       , ( I_PRICE * I_QTY )
                       , OL_DIST_INFO
                       , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                   FROM ( SELECT   :next_o_id as O_ID
                              , :w_id AS W_ID
                              , :d_id as D_ID
                              , OL_NUMBER
                              , I_ID
                              , I_QTY

                          FROM  Table( VALUES

                                  ( SMALLINT( 1 ) , :id0  , :ol_quantity0  )
                                , ( SMALLINT( 2 ) , :id1  , :ol_quantity1  )
                                , ( SMALLINT( 3 ) , :id2  , :ol_quantity2  )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                        ) AS ITEMLIST

                    , TABLE( NEW_OL_LOCAL(   I_ID
                                  , I_QTY
                                  , W_ID
                                  , O_ID
                                  , D_ID
                                  , OL_NUMBER
                                  )
                          ) AS NEW_OL_LOCAL

                  WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Local_4  CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                    , I_NAME     CHAR(24)
                    , I_DATA     VARCHAR(50)
                    , S_DATA     VARCHAR(50)
```

```
                    , S_QUANTITY SMALLINT )

                SELECT   O_ID
                    , D_ID
                    , W_ID
                    , OL_NUMBER
                    , I_ID
                    , W_ID AS I_SUPPLY_W_ID
                    , 0 AS OL_DELIVERY_D
                    , I_QTY
                    , ( I_PRICE * I_QTY )
                    , OL_DIST_INFO
                    , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                FROM ( SELECT   :next_o_id as O_ID
                           , :w_id AS W_ID
                           , :d_id as D_ID
                           , OL_NUMBER
                           , I_ID
                           , I_QTY

                       FROM  Table( VALUES

                                  (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                     ) AS ITEMLIST

                   , TABLE( NEW_OL_LOCAL(   I_ID
                                , I_QTY
                                , W_ID
                                , O_ID
                                , D_ID
                                , OL_NUMBER
                                )
                     ) AS NEW_OL_LOCAL

                WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Local_5  CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                    , I_NAME     CHAR(24)
                    , I_DATA     VARCHAR(50)
                    , S_DATA     VARCHAR(50)
                    , S_QUANTITY SMALLINT )


                SELECT   O_ID
                    , D_ID
                    , W_ID
                    , OL_NUMBER
                    , I_ID
                    , W_ID AS I_SUPPLY_W_ID
                    , 0 AS OL_DELIVERY_D
                    , I_QTY
                    , ( I_PRICE * I_QTY )
                    , OL_DIST_INFO
                    , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                FROM ( SELECT   :next_o_id as O_ID
                           , :w_id AS W_ID
                           , :d_id as D_ID
                           , OL_NUMBER
                           , I_ID
                           , I_QTY

                       FROM  Table( VALUES

                                  (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )
                                , (  SMALLINT( 5 )  , :id4  , :ol_quantity4  )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                     ) AS ITEMLIST

                   , TABLE( NEW_OL_LOCAL(   I_ID
                                , I_QTY
                                , W_ID
                                , O_ID
                                , D_ID
                                , OL_NUMBER
                                )
                     ) AS NEW_OL_LOCAL

                WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Local_6  CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                    , I_NAME     CHAR(24)
                    , I_DATA     VARCHAR(50)
                    , S_DATA     VARCHAR(50)
                    , S_QUANTITY SMALLINT )


                SELECT   O_ID
                    , D_ID
                    , W_ID
                    , OL_NUMBER
                    , I_ID
                    , W_ID AS I_SUPPLY_W_ID
                    , 0 AS OL_DELIVERY_D
                    , I_QTY
                    , ( I_PRICE * I_QTY )
                    , OL_DIST_INFO
                    , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                FROM ( SELECT   :next_o_id as O_ID
                           , :w_id AS W_ID
                           , :d_id as D_ID
                           , OL_NUMBER
                           , I_ID
                           , I_QTY

                       FROM  Table( VALUES

                                  (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )
                                , (  SMALLINT( 5 )  , :id4  , :ol_quantity4  )
                                , (  SMALLINT( 6 )  , :id5  , :ol_quantity5  )

                                ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                     ) AS ITEMLIST

                   , TABLE( NEW_OL_LOCAL(   I_ID
                                , I_QTY
                                , W_ID
                                , O_ID
                                , D_ID
                                , OL_NUMBER
                                )
                     ) AS NEW_OL_LOCAL

                WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Local_7  CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
```

```
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
               )

               INCLUDE (  I_PRICE   INTEGER
                     , I_NAME    CHAR(24)
                     , I_DATA    VARCHAR(50)
                     , S_DATA    VARCHAR(50)
                     , S_QUANTITY SMALLINT )


                    SELECT  O_ID
                         , D_ID
                         , W_ID
                         , OL_NUMBER
                         , I_ID
                         , W_ID AS I_SUPPLY_W_ID
                         , 0 AS OL_DELIVERY_D
                         , I_QTY
                         , ( I_PRICE * I_QTY )
                         , OL_DIST_INFO
                         , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT  :next_o_id as O_ID
                                 , :w_id AS W_ID
                                 , :d_id as D_ID
                                 , OL_NUMBER
                                 , I_ID
                                 , I_QTY

                              FROM  Table( VALUES

                                    (  SMALLINT( 1 ) , :id0  , :ol_quantity0  )
                                  , (  SMALLINT( 2 ) , :id1  , :ol_quantity1  )
                                  , (  SMALLINT( 3 ) , :id2  , :ol_quantity2  )
                                  , (  SMALLINT( 4 ) , :id3  , :ol_quantity3  )
                                  , (  SMALLINT( 5 ) , :id4  , :ol_quantity4  )
                                  , (  SMALLINT( 6 ) , :id5  , :ol_quantity5  )
                                  , (  SMALLINT( 7 ) , :id6  , :ol_quantity6  )

                                    ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                             ) AS ITEMLIST

                        , TABLE( NEW_OL_LOCAL(  I_ID
                                      , I_QTY
                                      , W_ID
                                      , O_ID
                                      , D_ID
                                      , OL_NUMBER
                                     )
                           ) AS NEW_OL_LOCAL

                       WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

                ) AS INS
;

EXEC SQL DECLARE ISOL_Local_8  CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

               (  OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
               )

               INCLUDE (  I_PRICE   INTEGER
                     , I_NAME    CHAR(24)
                     , I_DATA    VARCHAR(50)
                     , S_DATA    VARCHAR(50)
                     , S_QUANTITY SMALLINT )


                    SELECT  O_ID
                         , D_ID
                         , W_ID
                         , OL_NUMBER
                         , I_ID
                         , W_ID AS I_SUPPLY_W_ID
                         , 0 AS OL_DELIVERY_D
                         , I_QTY
                         , ( I_PRICE * I_QTY )
                         , OL_DIST_INFO
                         , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT  :next_o_id as O_ID
                                 , :w_id AS W_ID
                                 , :d_id as D_ID
                                 , OL_NUMBER
                                 , I_ID
                                 , I_QTY

                              FROM  Table( VALUES

                                    (  SMALLINT( 1 ) , :id0  , :ol_quantity0  )
                                  , (  SMALLINT( 2 ) , :id1  , :ol_quantity1  )
                                  , (  SMALLINT( 3 ) , :id2  , :ol_quantity2  )
                                  , (  SMALLINT( 4 ) , :id3  , :ol_quantity3  )
                                  , (  SMALLINT( 5 ) , :id4  , :ol_quantity4  )
                                  , (  SMALLINT( 6 ) , :id5  , :ol_quantity5  )
                                  , (  SMALLINT( 7 ) , :id6  , :ol_quantity6  )
                                  , (  SMALLINT( 8 ) , :id7  , :ol_quantity7  )

                                    ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                             ) AS ITEMLIST

                        , TABLE( NEW_OL_LOCAL(  I_ID
                                      , I_QTY
                                      , W_ID
                                      , O_ID
                                      , D_ID
                                      , OL_NUMBER
                                     )
                           ) AS NEW_OL_LOCAL

                       WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

                ) AS INS
;

EXEC SQL DECLARE ISOL_Local_9  CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

               (  OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
               )

               INCLUDE (  I_PRICE   INTEGER
                     , I_NAME    CHAR(24)
                     , I_DATA    VARCHAR(50)
                     , S_DATA    VARCHAR(50)
                     , S_QUANTITY SMALLINT )


                    SELECT  O_ID
                         , D_ID
                         , W_ID
                         , OL_NUMBER
                         , I_ID
                         , W_ID AS I_SUPPLY_W_ID
                         , 0 AS OL_DELIVERY_D
                         , I_QTY
                         , ( I_PRICE * I_QTY )
                         , OL_DIST_INFO
                         , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                        FROM ( SELECT  :next_o_id as O_ID
                                 , :w_id AS W_ID
                                 , :d_id as D_ID
                                 , OL_NUMBER
                                 , I_ID
                                 , I_QTY

                              FROM  Table( VALUES

                                    (  SMALLINT( 1 ) , :id0  , :ol_quantity0  )
                                  , (  SMALLINT( 2 ) , :id1  , :ol_quantity1  )
                                  , (  SMALLINT( 3 ) , :id2  , :ol_quantity2  )
                                  , (  SMALLINT( 4 ) , :id3  , :ol_quantity3  )
                                  , (  SMALLINT( 5 ) , :id4  , :ol_quantity4  )
                                  , (  SMALLINT( 6 ) , :id5  , :ol_quantity5  )
                                  , (  SMALLINT( 7 ) , :id6  , :ol_quantity6  )
                                  , (  SMALLINT( 8 ) , :id7  , :ol_quantity7  )
                                  , (  SMALLINT( 9 ) , :id8  , :ol_quantity8  )

                                    ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                             ) AS ITEMLIST

                        , TABLE( NEW_OL_LOCAL(  I_ID
                                      , I_QTY
                                      , W_ID
                                      , O_ID
                                      , D_ID
                                      , OL_NUMBER
                                     )
                           ) AS NEW_OL_LOCAL
```

```
                    WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Local_10 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR(50)
                        , S_DATA     VARCHAR(50)
                        , S_QUANTITY SMALLINT )


                     SELECT   O_ID
                          , D_ID
                          , W_ID
                          , OL_NUMBER
                          , I_ID
                          , W_ID AS I_SUPPLY_W_ID
                          , 0 AS OL_DELIVERY_D
                          , I_QTY
                          , ( I_PRICE * I_QTY )
                          , OL_DIST_INFO
                          , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                       FROM ( SELECT   :next_o_id as O_ID
                                  , :w_id AS W_ID
                                  , :d_id as D_ID
                                  , OL_NUMBER
                                  , I_ID
                                  , I_QTY

                              FROM  Table( VALUES

                                     (  SMALLINT( 1 ) , :id0  , :ol_quantity0  )
                                   , (  SMALLINT( 2 ) , :id1  , :ol_quantity1  )
                                   , (  SMALLINT( 3 ) , :id2  , :ol_quantity2  )
                                   , (  SMALLINT( 4 ) , :id3  , :ol_quantity3  )
                                   , (  SMALLINT( 5 ) , :id4  , :ol_quantity4  )
                                   , (  SMALLINT( 6 ) , :id5  , :ol_quantity5  )
                                   , (  SMALLINT( 7 ) , :id6  , :ol_quantity6  )
                                   , (  SMALLINT( 8 ) , :id7  , :ol_quantity7  )
                                   , (  SMALLINT( 9 ) , :id8  , :ol_quantity8  )
                                   , (  SMALLINT( 10 ) , :id9  , :ol_quantity9  )

                                  ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                            ) AS ITEMLIST

                       , TABLE( NEW_OL_LOCAL(   I_ID
                                              , I_QTY
                                              , W_ID
                                              , O_ID
                                              , D_ID
                                              , OL_NUMBER
                                              )
                              ) AS NEW_OL_LOCAL

                     WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Local_11 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR(50)
                        , S_DATA     VARCHAR(50)
                        , S_QUANTITY SMALLINT )


                     SELECT   O_ID
                          , D_ID
                          , W_ID
                          , OL_NUMBER
                          , I_ID
                          , W_ID AS I_SUPPLY_W_ID
                          , 0 AS OL_DELIVERY_D
                          , I_QTY
                          , ( I_PRICE * I_QTY )
                          , OL_DIST_INFO
                          , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                       FROM ( SELECT   :next_o_id as O_ID
                                  , :w_id AS W_ID
                                  , :d_id as D_ID
                                  , OL_NUMBER
                                  , I_ID
                                  , I_QTY

                              FROM  Table( VALUES

                                     (  SMALLINT( 1 ) , :id0  , :ol_quantity0  )
                                   , (  SMALLINT( 2 ) , :id1  , :ol_quantity1  )
                                   , (  SMALLINT( 3 ) , :id2  , :ol_quantity2  )
                                   , (  SMALLINT( 4 ) , :id3  , :ol_quantity3  )
                                   , (  SMALLINT( 5 ) , :id4  , :ol_quantity4  )
                                   , (  SMALLINT( 6 ) , :id5  , :ol_quantity5  )
                                   , (  SMALLINT( 7 ) , :id6  , :ol_quantity6  )
                                   , (  SMALLINT( 8 ) , :id7  , :ol_quantity7  )
                                   , (  SMALLINT( 9 ) , :id8  , :ol_quantity8  )
                                   , (  SMALLINT( 10 ) , :id9  , :ol_quantity9  )
                                   , (  SMALLINT( 11 ) , :id10 , :ol_quantity10 )

                                  ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                            ) AS ITEMLIST

                       , TABLE( NEW_OL_LOCAL(   I_ID
                                              , I_QTY
                                              , W_ID
                                              , O_ID
                                              , D_ID
                                              , OL_NUMBER
                                              )
                              ) AS NEW_OL_LOCAL

                     WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Local_12 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (   OL_O_ID
                , OL_D_ID
                , OL_W_ID
                , OL_NUMBER
                , OL_I_ID
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
              )

              INCLUDE (   I_PRICE    INTEGER
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR(50)
                        , S_DATA     VARCHAR(50)
                        , S_QUANTITY SMALLINT )


                     SELECT   O_ID
                          , D_ID
                          , W_ID
                          , OL_NUMBER
                          , I_ID
                          , W_ID AS I_SUPPLY_W_ID
                          , 0 AS OL_DELIVERY_D
                          , I_QTY
                          , ( I_PRICE * I_QTY )
                          , OL_DIST_INFO
                          , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                       FROM ( SELECT   :next_o_id as O_ID
                                  , :w_id AS W_ID
                                  , :d_id as D_ID
                                  , OL_NUMBER
                                  , I_ID
                                  , I_QTY
```

```
                        FROM  Table( VALUES

                                ( SMALLINT( 1 ) , :id0  , :ol_quantity0  )
                              , ( SMALLINT( 2 ) , :id1  , :ol_quantity1  )
                              , ( SMALLINT( 3 ) , :id2  , :ol_quantity2  )
                              , ( SMALLINT( 4 ) , :id3  , :ol_quantity3  )
                              , ( SMALLINT( 5 ) , :id4  , :ol_quantity4  )
                              , ( SMALLINT( 6 ) , :id5  , :ol_quantity5  )
                              , ( SMALLINT( 7 ) , :id6  , :ol_quantity6  )
                              , ( SMALLINT( 8 ) , :id7  , :ol_quantity7  )
                              , ( SMALLINT( 9 ) , :id8  , :ol_quantity8  )
                              , ( SMALLINT( 10 ) , :id9  , :ol_quantity9  )
                              , ( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
                              , ( SMALLINT( 12 ) , :id11 , :ol_quantity11 )

                             ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                      ) AS ITEMLIST

                    , TABLE( NEW_OL_LOCAL(  I_ID
                                          , I_QTY
                                          , W_ID
                                          , O_ID
                                          , D_ID
                                          , OL_NUMBER
                                          )
                      ) AS NEW_OL_LOCAL

                  WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

            ) AS INS
;

EXEC SQL DECLARE ISOL_Local_13 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (  OL_O_ID
               , OL_D_ID
               , OL_W_ID
               , OL_NUMBER
               , OL_I_ID
               , OL_SUPPLY_W_ID
               , OL_DELIVERY_D
               , OL_QUANTITY
               , OL_AMOUNT
               , OL_DIST_INFO
              )

              INCLUDE (  I_PRICE   INTEGER
                       , I_NAME    CHAR(24)
                       , I_DATA    VARCHAR(50)
                       , S_DATA    VARCHAR(50)
                       , S_QUANTITY SMALLINT )


                   SELECT  O_ID
                        , D_ID
                        , W_ID
                        , OL_NUMBER
                        , I_ID
                        , W_ID AS I_SUPPLY_W_ID
                        , 0 AS OL_DELIVERY_D
                        , I_QTY
                        , ( I_PRICE * I_QTY )
                        , OL_DIST_INFO
                        , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                    FROM ( SELECT  :next_o_id as O_ID
                                 , :w_id AS W_ID
                                 , :d_id as D_ID
                                 , OL_NUMBER
                                 , I_ID
                                 , I_QTY

                             FROM  Table( VALUES

                                     ( SMALLINT( 1 ) , :id0  , :ol_quantity0  )
                                   , ( SMALLINT( 2 ) , :id1  , :ol_quantity1  )
                                   , ( SMALLINT( 3 ) , :id2  , :ol_quantity2  )
                                   , ( SMALLINT( 4 ) , :id3  , :ol_quantity3  )
                                   , ( SMALLINT( 5 ) , :id4  , :ol_quantity4  )
                                   , ( SMALLINT( 6 ) , :id5  , :ol_quantity5  )
                                   , ( SMALLINT( 7 ) , :id6  , :ol_quantity6  )
                                   , ( SMALLINT( 8 ) , :id7  , :ol_quantity7  )
                                   , ( SMALLINT( 9 ) , :id8  , :ol_quantity8  )
                                   , ( SMALLINT( 10 ) , :id9  , :ol_quantity9  )
                                   , ( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
                                   , ( SMALLINT( 12 ) , :id11 , :ol_quantity11 )
                                   , ( SMALLINT( 13 ) , :id12 , :ol_quantity12 )

                                  ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                           ) AS ITEMLIST

                         , TABLE( NEW_OL_LOCAL(  I_ID
                                               , I_QTY
                                               , W_ID
                                               , O_ID
                                               , D_ID
                                               , OL_NUMBER
                                               )
                           ) AS NEW_OL_LOCAL

                       WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

                 ) AS INS
;

EXEC SQL DECLARE ISOL_Local_14 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (  OL_O_ID
               , OL_D_ID
               , OL_W_ID
               , OL_NUMBER
               , OL_I_ID
               , OL_SUPPLY_W_ID
               , OL_DELIVERY_D
               , OL_QUANTITY
               , OL_AMOUNT
               , OL_DIST_INFO
              )

              INCLUDE (  I_PRICE   INTEGER
                       , I_NAME    CHAR(24)
                       , I_DATA    VARCHAR(50)
                       , S_DATA    VARCHAR(50)
                       , S_QUANTITY SMALLINT )


                   SELECT  O_ID
                        , D_ID
                        , W_ID
                        , OL_NUMBER
                        , I_ID
                        , W_ID AS I_SUPPLY_W_ID
                        , 0 AS OL_DELIVERY_D
                        , I_QTY
                        , ( I_PRICE * I_QTY )
                        , OL_DIST_INFO
                        , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                    FROM ( SELECT  :next_o_id as O_ID
                                 , :w_id AS W_ID
                                 , :d_id as D_ID
                                 , OL_NUMBER
                                 , I_ID
                                 , I_QTY

                             FROM  Table( VALUES

                                     ( SMALLINT( 1 ) , :id0  , :ol_quantity0  )
                                   , ( SMALLINT( 2 ) , :id1  , :ol_quantity1  )
                                   , ( SMALLINT( 3 ) , :id2  , :ol_quantity2  )
                                   , ( SMALLINT( 4 ) , :id3  , :ol_quantity3  )
                                   , ( SMALLINT( 5 ) , :id4  , :ol_quantity4  )
                                   , ( SMALLINT( 6 ) , :id5  , :ol_quantity5  )
                                   , ( SMALLINT( 7 ) , :id6  , :ol_quantity6  )
                                   , ( SMALLINT( 8 ) , :id7  , :ol_quantity7  )
                                   , ( SMALLINT( 9 ) , :id8  , :ol_quantity8  )
                                   , ( SMALLINT( 10 ) , :id9  , :ol_quantity9  )
                                   , ( SMALLINT( 11 ) , :id10 , :ol_quantity10 )
                                   , ( SMALLINT( 12 ) , :id11 , :ol_quantity11 )
                                   , ( SMALLINT( 13 ) , :id12 , :ol_quantity12 )
                                   , ( SMALLINT( 14 ) , :id13 , :ol_quantity13 )

                                  ) AS X ( OL_NUMBER , I_ID  , I_QTY          )
                           ) AS ITEMLIST

                         , TABLE( NEW_OL_LOCAL(  I_ID
                                               , I_QTY
                                               , W_ID
                                               , O_ID
                                               , D_ID
                                               , OL_NUMBER
                                               )
                           ) AS NEW_OL_LOCAL

                       WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

                 ) AS INS
;

EXEC SQL DECLARE ISOL_Local_15 CURSOR FOR

  SELECT I_PRICE , I_NAME , I_DATA , OL_DIST_INFO , S_DATA , S_QUANTITY

    FROM NEW TABLE ( INSERT INTO ORDER_LINE

              (  OL_O_ID
               , OL_D_ID
               , OL_W_ID
               , OL_NUMBER
               , OL_I_ID
```

```
                , OL_SUPPLY_W_ID
                , OL_DELIVERY_D
                , OL_QUANTITY
                , OL_AMOUNT
                , OL_DIST_INFO
               )

              INCLUDE (   I_PRICE    INTEGER
                        , I_NAME     CHAR(24)
                        , I_DATA     VARCHAR(50)
                        , S_DATA     VARCHAR(50)
                        , S_QUANTITY SMALLINT )


                   SELECT   O_ID
                          , D_ID
                          , W_ID
                          , OL_NUMBER
                          , I_ID
                          , W_ID AS I_SUPPLY_W_ID
                          , 0 AS OL_DELIVERY_D
                          , I_QTY
                          , ( I_PRICE * I_QTY )
                          , OL_DIST_INFO
                          , I_PRICE, I_NAME, I_DATA, S_DATA, S_QUANTITY

                    FROM ( SELECT   :next_o_id as O_ID
                                  , :w_id AS W_ID
                                  , :d_id as D_ID
                                  , OL_NUMBER
                                  , I_ID
                                  , I_QTY

                             FROM  Table( VALUES

                                        (  SMALLINT( 1 )  , :id0  , :ol_quantity0  )
                                      , (  SMALLINT( 2 )  , :id1  , :ol_quantity1  )
                                      , (  SMALLINT( 3 )  , :id2  , :ol_quantity2  )
                                      , (  SMALLINT( 4 )  , :id3  , :ol_quantity3  )
                                      , (  SMALLINT( 5 )  , :id4  , :ol_quantity4  )
                                      , (  SMALLINT( 6 )  , :id5  , :ol_quantity5  )
                                      , (  SMALLINT( 7 )  , :id6  , :ol_quantity6  )
                                      , (  SMALLINT( 8 )  , :id7  , :ol_quantity7  )
                                      , (  SMALLINT( 9 )  , :id8  , :ol_quantity8  )
                                      , (  SMALLINT( 10 ) , :id9  , :ol_quantity9  )
                                      , (  SMALLINT( 11 ) , :id10 , :ol_quantity10 )
                                      , (  SMALLINT( 12 ) , :id11 , :ol_quantity11 )
                                      , (  SMALLINT( 13 ) , :id12 , :ol_quantity12 )
                                      , (  SMALLINT( 14 ) , :id13 , :ol_quantity13 )
                                      , (  SMALLINT( 15 ) , :id14 , :ol_quantity14 )

                                    ) AS X ( OL_NUMBER , I_ID  , I_QTY           )
                           ) AS ITEMLIST

                    , TABLE( NEW_OL_LOCAL(   I_ID
                                           , I_QTY
                                           , W_ID
                                           , O_ID
                                           , D_ID
                                           , OL_NUMBER
                                           )
                           ) AS NEW_OL_LOCAL

                    WHERE NEW_OL_LOCAL.I_PRICE IS NOT NULL

               ) AS INS
;

 in_neword = (struct in_neword_struct *)  pin ;
 neword    = (struct out_neword_struct *) pout ;

 for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < in_neword->s_O_OL_CNT ;
inputItemArrayIndex++ )
 {
   i_priceArray[ inputItemArrayIndex ] = 0 ;
 }

 neword->deadlocks = -1 ;

retry_tran:

 neword->deadlocks++ ;

 EXEC SQL

   SELECT D_TAX, D_NEXT_O_ID  INTO :dist_tax , :next_o_id

     FROM OLD TABLE ( UPDATE DISTRICT

                        SET  D_NEXT_O_ID = D_NEXT_O_ID + 1

                      WHERE D_W_ID = :w_id
                        AND D_ID = :d_id
               ) AS OT
;

 if ( sqlca.sqlcode != 0 )
 {
   DLCHK( retry_tran );
   sqlerror( NEWORD_SQL, "DISTRICT", __LINE__, &sqlca ) ;
   goto ferror;
 }

 #define NEW_CURSOR_OPEN_ERROR                                          \
 {                                                                      \
   if( sqlca.sqlcode != 0    )                                          \
   {                                                                    \
     goto sql_error ;                                                   \
   }                                                                    \
 }

 #define NEW_CURSOR_ERROR                                               \
 {                                                                      \
   if( sqlca.sqlcode == 0    )                                          \
   {                                                                    \
     neword->s_O_OL_CNT ++ ;                                            \
   }                                                                    \
   else                                                                 \
   if( sqlca.sqlcode == +100 )                                          \
   {                                                                    \
     break ;                                                            \
   }                                                                    \
   else                                                                 \
     goto sql_error ;                                                   \
 }

 if ( allLocal )
 {
   switch( inputItemCount )
   {
     case 1:
       EXEC SQL OPEN ISOL_Local_1 ;
       NEW_CURSOR_OPEN_ERROR
       for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
       {
         EXEC SQL FETCH ISOL_Local_1  INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
         NEW_CURSOR_ERROR
       }
       break ;
     case 2:
       EXEC SQL OPEN ISOL_Local_2 ;
       NEW_CURSOR_OPEN_ERROR
       for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
       {
         EXEC SQL FETCH ISOL_Local_2  INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
         NEW_CURSOR_ERROR
       }
       break ;
     case 3:
       EXEC SQL OPEN ISOL_Local_3 ;
       NEW_CURSOR_OPEN_ERROR
       for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
       {
         EXEC SQL FETCH ISOL_Local_3  INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
         NEW_CURSOR_ERROR
       }
       break ;
     case 4:
       EXEC SQL OPEN ISOL_Local_4 ;
       NEW_CURSOR_OPEN_ERROR
       for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
       {
         EXEC SQL FETCH ISOL_Local_4  INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
         NEW_CURSOR_ERROR
       }
       break ;
     case 5:
       EXEC SQL OPEN ISOL_Local_5 ;
       NEW_CURSOR_OPEN_ERROR
       for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
       {
         EXEC SQL FETCH ISOL_Local_5  INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
         NEW_CURSOR_ERROR
       }
       break ;
     case 6:
       EXEC SQL OPEN ISOL_Local_6 ;
       NEW_CURSOR_OPEN_ERROR
       for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
       {
         EXEC SQL FETCH ISOL_Local_6  INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
         NEW_CURSOR_ERROR
       }
       break ;
     case 7:
       EXEC SQL OPEN ISOL_Local_7 ;
       NEW_CURSOR_OPEN_ERROR
       for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
       {
         EXEC SQL FETCH ISOL_Local_7  INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
         NEW_CURSOR_ERROR
```

```
        }
        break ;
      case 8:
        EXEC SQL OPEN ISOL_Local_8 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_8 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 9:
        EXEC SQL OPEN ISOL_Local_9 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_9 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 10:
        EXEC SQL OPEN ISOL_Local_10 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_10 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 11:
        EXEC SQL OPEN ISOL_Local_11 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_11 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 12:
        EXEC SQL OPEN ISOL_Local_12 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_12 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 13:
        EXEC SQL OPEN ISOL_Local_13 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_13 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 14:
        EXEC SQL OPEN ISOL_Local_14 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_14 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 15:
        EXEC SQL OPEN ISOL_Local_15 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Local_15 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;

      default:
        sqlerror(NEWORD_SQL, "Default switch on local  orderline/stock/index", __LINE__, &sqlca );
        goto ferror;
    }
  }
  else
  {
    switch( inputItemCount )
    {
      case 1:
        EXEC SQL OPEN ISOL_Remote_1 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Remote_1 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 2:
        EXEC SQL OPEN ISOL_Remote_2 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Remote_2 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 3:
        EXEC SQL OPEN ISOL_Remote_3 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Remote_3 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 4:
        EXEC SQL OPEN ISOL_Remote_4 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Remote_4 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 5:
        EXEC SQL OPEN ISOL_Remote_5 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Remote_5 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 6:
        EXEC SQL OPEN ISOL_Remote_6 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Remote_6 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 7:
        EXEC SQL OPEN ISOL_Remote_7 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Remote_7 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 8:
        EXEC SQL OPEN ISOL_Remote_8 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Remote_8 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 9:
        EXEC SQL OPEN ISOL_Remote_9 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Remote_9 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 10:
        EXEC SQL OPEN ISOL_Remote_10 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Remote_10 INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
          NEW_CURSOR_ERROR
        }
        break ;
      case 11:
        EXEC SQL OPEN ISOL_Remote_11 ;
        NEW_CURSOR_OPEN_ERROR
        for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
        {
          EXEC SQL FETCH ISOL_Remote_11 INTO :item_price, :item_name, :i_data,
```

```
:stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 12:
          EXEC SQL OPEN ISOL_Remote_12 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_12  INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 13:
          EXEC SQL OPEN ISOL_Remote_13 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_13  INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 14:
          EXEC SQL OPEN ISOL_Remote_14 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_14  INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;
        case 15:
          EXEC SQL OPEN ISOL_Remote_15 ;
          NEW_CURSOR_OPEN_ERROR
          for ( inputItemArrayIndex = 0 ; inputItemArrayIndex < inputItemCount ; inputItemArrayIndex++ )
          {
            EXEC SQL FETCH ISOL_Remote_15  INTO :item_price, :item_name, :i_data,
:stockDistrictInformation , :s_data , :s_quantity ;
            NEW_CURSOR_ERROR
          }
          break ;

        default:
          sqlerror(NEWORD_SQL, "Default switch on remote orderline/stock/index", __LINE__, &sqlca);
          goto ferror;
      }
    }

    for ( inputItemArrayIndex = 0 ;
          inputItemArrayIndex < in_neword->s_O_OL_CNT
        && i_priceArray[ inputItemArrayIndex ] != 0 ;
        inputItemArrayIndex++ )
    {
      neword->item[ inputItemArrayIndex ].s_I_PRICE = i_priceArray[ inputItemArrayIndex ] ;

      if (    is_ORIGINAL( s_dataArray[ inputItemArrayIndex ].data, s_dataArray[ inputItemArrayIndex ].len
)
           && is_ORIGINAL( i_dataArray[ inputItemArrayIndex ].data, i_dataArray[ inputItemArrayIndex
].len )    )
      {
        neword->item[ inputItemArrayIndex ].s_brand_generic = 'B';
      }
      else
      {
        neword->item[ inputItemArrayIndex ].s_brand_generic = 'G';
      }
    }

    EXEC SQL

      SELECT W_TAX, C_DISCOUNT, C_LAST, C_CREDIT

        INTO :ware_tax, :c_discount, :c_last, :c_credit

        FROM TABLE ( NEW_WH (   :next_o_id
                              , :w_id
                              , :d_id
                              , :c_id
                              , :o_entry_d
                              , :inputItemCount
                              , :allLocal
                            )
                   ) AS NEW_WH_TABLE
    ;

    if ( sqlca.sqlcode == 0 )
    {
      if ( neword->s_O_OL_CNT == in_neword->s_O_OL_CNT  )
      {
        EXEC SQL COMMIT;

        if( sqlca.sqlcode == 0 )
        {
          neword->s_transtatus = TRAN_OK;
        }
        else
        {
          sqlerror(NEWORD_SQL, "COMMIT", __LINE__ , &sqlca ) ;
          goto ferror;
        }
      }
      else
      {
        neword->s_transtatus = INVALID_ITEM ;

        EXEC SQL ROLLBACK WORK ;

        if ( sqlca.sqlcode != 0 )
        {
          neword->s_transtatus = FATAL_SQLERROR;

          sqlerror(NEWORD_SQL, "ROLLBACK", __LINE__ , &sqlca);
        }
      }
    }
    else
    {
      DLCHK( retry_tran );

      sqlerror( NEWORD_SQL, "NEW_WH", __LINE__ , &sqlca);
      goto ferror;
    }

    /*-----------------------------------------------*/
    /*  Return to client                             */
    /*-----------------------------------------------*/

  mexit:

    if ( sqlca.sqlcode >= 0 )
    {
      rc = SQLZ_HOLD_PROC ;
    }
    else
    {
      rc = SQLZ_DISCONNECT_PROC ;
    }

    return( rc ) ;

  sql_error:
    {
      char tempstr[ 4096 ] ;

      DLCHK( retry_tran ) ;

      sprintf( tempstr, "inputItemCount=%d, :next_o_id=%d, :d_id=%d, :w_id=%d", inputItemCount,
next_o_id, d_id, w_id ) ;
      sqlerror( NEWORD_SQL, tempstr , __LINE__, &sqlca ) ;
    }

  ferror:

    neword->s_transtatus = FATAL_SQLERROR;

    EXEC SQL rollback work;

    if ( sqlca.sqlcode != 0 )
    {
      sqlerror( NEWORD_SQL, "ROLLBACK FAILED", __LINE__ , &sqlca ) ;
    }

    new_debug( neword, in_neword, "ferror code" ) ;

    goto mexit ;

  }

  /*
  ** A little function to search for the string "ORIGINAL" given a string and
  ** it's length
  */
  static unsigned char  skip[256] = {8,8,8,8,8,8,8,8,8,8,  /*0-9*/
                                     8,8,8,8,8,8,8,8,8,8,  /*10-19*/
                                     8,8,8,8,8,8,8,8,8,8,  /*20-29*/
                                     8,8,8,8,8,8,8,8,8,8,  /*30-39*/
                                     8,8,8,8,8,8,8,8,8,8,  /*40-49*/
                                     8,8,8,8,8,8,8,8,8,8,  /*50-59*/
                                     8,8,8,8,8,1,8,8,8,8,  /*60-69*/
                                     8,4,8,3,8,8,0,8,2,7,  /*70-79*/
                                     8,8,6,8,8,8,8,8,8,8,  /*80-89*/
                                     8,8,8,8,8,8,8,8,8,8,  /*90-99*/
                                     8,8,8,8,8,8,8,8,8,8,  /*100-109*/
                                     8,8,8,8,8,8,8,8,8,8,  /*110-119*/
                                     8,8,8,8,8,8,8,8,8,8,  /*120-129*/
                                     8,8,8,8,8,8,8,8,8,8,  /*130-139*/
                                     8,8,8,8,8,8,8,8,8,8,  /*140-149*/
                                     8,8,8,8,8,8,8,8,8,8,  /*150-159*/
                                     8,8,8,8,8,8,8,8,8,8,  /*160-169*/
                                     8,8,8,8,8,8,8,8,8,8,  /*170-179*/
                                     8,8,8,8,8,8,8,8,8,8,  /*180-189*/
                                     8,8,8,8,8,8,8,8,8,8,  /*190-199*/
                                     8,8,8,8,8,8,8,8,8,8,  /*200-209*/
                                     8,8,8,8,8,8,8,8,8,8,  /*210-219*/
                                     8,8,8,8,8,8,8,8,8,8,  /*220-229*/
```

```
                    8,8,8,8,8,8,8,8,8,8,  /*230-239*/
                    8,8,8,8,8,8,8,8,8,8,  /*240-249*/
                    8,8,8,8,8};           /*250-254*/

static int is_ORIGINAL( char *string, short length )
{
 char           *cur_string;
 char           *end_string;
 unsigned char   *skips;
 int             skip_dist;
 int             result = 0;

 cur_string = string+7;
 end_string = string + length;
 skips = skip;

 while (cur_string < end_string)
  {
   skip_dist = skips[*cur_string];
   while ( (skip_dist > 0) && (cur_string < end_string) )
    {
     skip_dist = skips[*(cur_string += skip_dist)];
    }

   if (cur_string >= end_string)
    goto exit;

   if ( cur_string[-4] != 'G' )
    goto noMatch;

   if ( memcmp( cur_string-7, "ORIGINAL", 8 ) == 0 )
    {
     result = 1;
     goto exit;
    }
noMatch:
   cur_string += 8;
  } /* end while */

exit:
 return( result );
}


// ---------------------------------------------------------------------
// Payment CLIENT
// ---------------------------------------------------------------------

#undef w_id
#undef d_id
#undef c_discount
#undef c_id
#undef c_last
#undef c_credit

int payment_sql (   struct in_payment_struct * in_payment
                  , struct out_payment_struct * payment    )
{
 struct sqlca sqlca ;

 int rc = 0 ;

 EXEC SQL BEGIN DECLARE SECTION;

   // Inputs

   sqlint64 h_amount ;
   sqlint32 in_c_id ;

   //char     c_last_input[ 16 ] ;

   struct s_data_type { short len ; char data[ 16 ] ; } c_last_input ;

   //##sqlint32 w_id ;
   sqlint32 c_w_id ;
   //##short    d_id ;
   short    c_d_id ;
   sqlint64 h_date;

   // Outputs

   //##sqlint32 c_id ;

   sqlint64  c_credit_lim ;
   //##double  c_discount ;
   sqlint64  c_balance ;

   char    w_street_1 [ 20 ] , w_street_2 [ 20 ] ;
   char    w_city [ 20 ] ,     w_state [ 2 ] ,      w_zip [ 9 ] ;

   char    d_street_1 [ 20 ] , d_street_2 [ 20 ] ,  d_city [ 20 ] ;
   char    d_state [ 2 ] ,     d_zip [ 9 ] ,        c_first [ 16 ] ;

   //##char    c_last [ 16 ] ;

   char    c_middle [ 2 ]    , c_street_1 [ 20 ] ;
   char    c_street_2 [ 20 ] , c_city [ 20 ] ,      c_state [ 2 ] ;
   char    c_zip [ 9 ] ,       c_phone [ 16 ] ;
   //##char    c_credit [ 2 ] ;

   sqlint64    c_since ;

   char    c_data [ 200 ] ;
   short   c_data_indicator = 0 ;

   struct c_data_prefix_c_last_type { short len ; char data[ 28 ] ; } c_data_prefix_c_last ;
   struct c_data_prefix_c_id_type   { short len ; char data[ 34 ] ; } c_data_prefix_c_id ;


 EXEC SQL END DECLARE SECTION;


 // Input redirects

 #define h_amount          in_payment->s_H_AMOUNT
 #define in_c_id           in_payment->s_C_ID

 #define w_id              in_payment->s_W_ID
 #define d_id              in_payment->s_D_ID

 #define c_d_id            in_payment->s_C_D_ID
 #define c_w_id            in_payment->s_C_W_ID
 #define h_date            in_payment->s_H_DATE_time

 // Output redirects

 #define c_credit_lim      payment->s_C_CREDIT_LIM
 #define c_discount        payment->s_C_DISCOUNT
 #define c_balance         payment->s_C_BALANCE

 #define c_id              payment->s_C_ID
 #define c_last            payment->s_C_LAST

 #define c_first           payment->s_C_FIRST
 #define c_middle          payment->s_C_MIDDLE
 #define c_street_1        payment->s_C_STREET_1
 #define c_street_2        payment->s_C_STREET_2
 #define c_city            payment->s_C_CITY
 #define c_state           payment->s_C_STATE
 #define c_zip             payment->s_C_ZIP
 #define c_phone           payment->s_C_PHONE
 #define c_credit          payment->s_C_CREDIT
 #define c_since           payment->s_C_SINCE_time
 #define c_data            payment->s_C_DATA

 #define w_street_1        payment->s_W_STREET_1
 #define w_street_2        payment->s_W_STREET_2
 #define w_city            payment->s_W_CITY
 #define w_state           payment->s_W_STATE
 #define w_zip             payment->s_W_ZIP

 #define d_street_1        payment->s_D_STREET_1
 #define d_street_2        payment->s_D_STREET_2
 #define d_city            payment->s_D_CITY
 #define d_state           payment->s_D_STATE
 #define d_zip             payment->s_D_ZIP

 /* Create Timestamp */

 in_payment->s_H_DATE_time = (sqlint64) time( NULL ) ;
 payment->s_H_DATE_time = in_payment->s_H_DATE_time ;

 payment->deadlocks = -1 ;

 if (c_w_id == 0) { c_w_id = w_id; }
 if (c_d_id == 0) { c_d_id = d_id; }

#ifdef DEBUGIT
  pay_debug(payment, in_payment, "Client before call");
#endif /* DEBUGIT */

 if ( in_c_id == 0 )
 {
   c_data_prefix_c_last.len = sprintf( c_data_prefix_c_last.data, " %2.2d %6.6d %2.2d %6.6d
%04.4d.%02.2d",  c_d_id , c_w_id , d_id , w_id , (int)(h_amount / 100) , (int)(h_amount % 100) ) ;

   c_last_input.len = strlen( in_payment->s_C_LAST ) ;
   memcpy( c_last_input.data , in_payment->s_C_LAST , c_last_input.len ) ;

   memcpy( c_last, c_last_input.data, sizeof( c_last ) ) ;

 } else {

   c_id = in_c_id ;

   c_data_prefix_c_id.len = sprintf( c_data_prefix_c_id.data, " %5.5d %2.2d %6.6d %2.2d %6.6d
%04.4d.%02.2d", c_id , c_d_id , c_w_id , d_id , w_id , (int)(h_amount / 100), (int)(h_amount % 100) ) ;

 }

retry_tran:

 payment->deadlocks ++ ;

 if ( in_c_id == 0 )
 {
     EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC
```

```
      SELECT   W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP
          , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
          , C_ID, C_FIRST, C_MIDDLE, 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

       INTO   :w_street_1 , :w_street_2 , :w_city , :w_state , :w_zip
          , :d_street_1 , :d_street_2 , :d_city , :d_state , :d_zip
          , :c_id , :c_first , :c_middle ,   :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 :c_data_indicator

       FROM TABLE ( PAY_C_LAST(   :w_id
                        , :d_id
                        , :c_w_id
                        , :c_d_id
                        , :c_last_input
                        , :h_date
                        , :h_amount
                        , :c_data_prefix_c_last
                       )

             ) AS T (   W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP
                    , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
                    , C_ID, C_FIRST, C_MIDDLE, 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
                   )
      ;

      COMMIT ;

    END COMPOUND ;

  }
  else
  {
      EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

      SELECT   W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP
          , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
          , C_LAST, C_FIRST, C_MIDDLE, 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

       INTO   :w_street_1 , :w_street_2 , :w_city , :w_state , :w_zip
          , :d_street_1 , :d_street_2 , :d_city , :d_state , :d_zip
          , :c_last, :c_first , :c_middle , :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 :c_data_indicator

       FROM TABLE ( PAY_C_ID(   :w_id
                        , :d_id
                        , :c_w_id
                        , :c_d_id
                        , :in_c_id
                        , :h_date
                        , :h_amount
                        , :c_data_prefix_c_id
                       )

             ) AS T(   W_STREET_1, W_STREET_2, W_CITY, W_STATE,  W_ZIP
                    , D_STREET_1, D_STREET_2, D_CITY, D_STATE, D_ZIP
                    , C_LAST, C_FIRST, C_MIDDLE, 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
                   )
      ;

      COMMIT ;

    END COMPOUND ;

  }

#ifdef DEBUGIT
  pay_debug(payment, in_payment, "Client after SP call");
#endif /* DEBUGIT */

  if ( sqlca.sqlcode == 0 )
  {
    payment->s_transtatus = TRAN_OK ;
  }
  else
  {
    DLCHK( retry_tran ) ;

    sqlerror( PAYMENT_SQL , "PAY" , __LINE__ , &sqlca);
    goto ferror ;
  }

mexit:

  if ( sqlca.sqlcode )
  {
    sqlerror( PAYMENT_SQL, "PAY", __LINE__, &sqlca) ;
    pay_debug( payment, in_payment, "sqlcode != 0" ) ;
    payment->s_transtatus = FATAL_SQLERROR;
    rc = -1 ;
  }

  if ( payment->s_transtatus <= FATAL_SQLERROR )
  {
    pay_debug(payment, in_payment, "s_transtatus <= FATAL_SQLERROR");
    rc = -1 ;
  }

  return rc ;

ferror:

  payment->s_transtatus = FATAL_SQLERROR ;

  EXEC SQL ROLLBACK WORK ;

  if ( sqlca.sqlcode != 0 )
  {
    sqlerror( PAYMENT_SQL, "ROLLBACK FAILED", __LINE__, &sqlca ) ;
  }

  goto mexit ;
}

// --------------------------------------------------------------------
// Order Status CLIENT
// --------------------------------------------------------------------


int ordstat_sql  (  struct in_ordstat_struct * in_ordstat
              , struct out_ordstat_struct * ordstat)
{
  struct sqlca sqlca ;

  EXEC SQL BEGIN DECLARE SECTION;

    struct vc_ord_in
    {
      short len ;
      char data[ 42 ] ;
    } * in_ord ;

    struct vc_ord_out
    {
      short len ;
      char data[ 446 ] ;
    } * out_ord ;

  EXEC SQL END DECLARE SECTION;

  int rc = 0 ;

  in_ord     = (struct vc_ord_in *) in_ordstat ;
  in_ord->len = sizeof(struct in_ordstat_struct) - SPGENERAL_ADJUST ;

  out_ord     = (struct vc_ord_out *) ordstat ;
  out_ord->len = sizeof(struct out_ordstat_struct) - SPGENERAL_ADJUST ;

#ifdef DEBUGIT
  ord_debug(ordstat, in_ordstat, "Client before SP call");
#endif /* DEBUGIT */

  EXEC SQL CALL ords ( :*in_ord, :*out_ord ) ;

#ifdef DEBUGIT
  ord_debug(ordstat, in_ordstat, "Client after SP call");
#endif /* DEBUGIT */

  if ( sqlca.sqlcode == 0 )
  {
    if ( in_ordstat->s_C_ID == 0 )
    {
      memcpy( ordstat->s_C_LAST , in_ordstat->s_C_LAST, sizeof( ordstat->s_C_LAST ) ) ;
    }
    else
    {
      ordstat->s_C_ID = in_ordstat->s_C_ID ;
    }
  }
  else
  {
    sqlerror( ORDSTAT_SQL, "ORD", __LINE__, &sqlca) ;
    ord_debug( ordstat, in_ordstat, "sqlcode != 0" ) ;
    ordstat->s_transtatus = FATAL_SQLERROR;
    rc = -1 ;
  }

  if ( ordstat->s_transtatus <= FATAL_SQLERROR )
  {
    ord_debug(ordstat, in_ordstat, "s_transtatus <= FATAL_SQLERROR");
    rc = -1 ;
  }

  return ( rc ) ;
}


// --------------------------------------------------------------------
```

```
// Order Status SERVER
// ---------------------------------------------------------------------

#undef w_id
#undef d_id
#undef o_entry_d
#undef c_id
#undef c_first
#undef c_middle
#undef c_last
#undef c_balance

SQL_API_RC order_status_internal( char *pin, char *pout )
{
  struct in_ordstat_struct  * in_ordstat = (struct in_ordstat_struct *) pin ;
  struct out_ordstat_struct * ordstat = (struct out_ordstat_struct *) pout ;

  struct sqlca sqlca ;

  EXEC SQL BEGIN DECLARE SECTION;

   // From input values

   //##sqlint32   w_id ;
   //##short      d_id;
   sqlint32   c_id_input ;

   //##struct s_data_type { short len ; char data[ 16 ] ; } c_last_input ;

   // From queries

   // From initial query

   sqlint32   o_id ;
   //##sqlint32   c_id ;
   short      o_carrier_id ;
   //##sqlint64   o_entry_d ;

   //##char   c_first[ 16 ] ;
   //##char   c_middle[ 2 ] ;
   //##char   c_last[ 16 ] ;
   //##double c_balance ;

   // From cursor

   sqlint32   ol_i_id ;
   sqlint32   ol_supply_w_id ;
   short      ol_quantity ;
   sqlint32   ol_amount ;
   sqlint64   ol_delivery_d ;

  EXEC SQL END DECLARE SECTION;

  struct s_data_type { short len ; char data[ 16 ] ; } c_last_input ;

  int     rc = 0 ;
  int     itemArrayIndex = 0 ;

  #define w_id               in_ordstat->s_W_ID ;
  #define d_id               in_ordstat->s_D_ID ;
  #define c_id_input         in_ordstat->s_C_ID

  //#define c_last_input       in_ordstat->s_C_LAST

  #define o_id               ordstat->s_O_ID
  #define o_entry_d          ordstat->s_O_ENTRY_D_time
  #define o_carrier_id       ordstat->s_O_CARRIER_ID
  #define c_id               ordstat->s_C_ID
  #define c_first            ordstat->s_C_FIRST
  #define c_middle           ordstat->s_C_MIDDLE
  #define c_last             ordstat->s_C_LAST
  #define c_balance          ordstat->s_C_BALANCE

  EXEC SQL DECLARE read_orderline_cur CURSOR FOR

    SELECT OL_I_ID, OL_SUPPLY_W_ID, OL_QUANTITY, OL_AMOUNT, OL_DELIVERY_D

      FROM ORDER_LINE

      WHERE OL_W_ID = :w_id
        AND OL_D_ID = :d_id
        AND OL_O_ID = :o_id

      FOR FETCH ONLY ;

  ordstat->deadlocks = -1 ;

retry_tran:

  ordstat->deadlocks ++ ;

  if ( c_id_input == 0 )
  {
    c_last_input.len = strlen( in_ordstat->s_C_LAST ) ;
    memcpy( c_last_input.data , in_ordstat->s_C_LAST , c_last_input.len ) ;

    EXEC SQL

      SELECT O_ID, O_CARRIER_ID, O_ENTRY_D, C_BALANCE, C_FIRST, C_MIDDLE, C_ID

          INTO :o_id, :o_carrier_id , :o_entry_d , :c_balance, :c_first, :c_middle, :c_id

          FROM TABLE ( ORD_C_LAST(   :w_id
                                   , :d_id
                                   , :c_last_input
                                 )
                     ) AS ORD_C_LAST
    ;
  }
  else
  {
    EXEC SQL

      SELECT O_ID, O_CARRIER_ID, O_ENTRY_D , C_BALANCE, C_FIRST, C_MIDDLE ,C_LAST

          INTO :o_id, :o_carrier_id , :o_entry_d , :c_balance, :c_first, :c_middle, :c_last

          FROM TABLE ( ORD_C_ID(   :w_id
                                 , :d_id
                                 , :c_id_input
                               )
                     ) AS ORD_C_ID
    ;
  }

  if ( sqlca.sqlcode != 0 )
  {
    DLCHK( retry_tran );
    sqlerror( ORDSTAT_SQL, "READ CUST and ORDERS", __LINE__ ,  &sqlca ) ;
    goto ferror;
  }

  /*-------------------------------------------------*/
  /*  Read ORDER_LINEs                               */
  /*-------------------------------------------------*/

  EXEC SQL OPEN read_orderline_cur ;

  if ( sqlca.sqlcode != 0 )
  {
    DLCHK( retry_tran );
    sqlerror(ORDSTAT_SQL, "OPEN CURSOR read_orderline_cur", __LINE__ , &sqlca ) ;
    goto ferror;
  }

  itemArrayIndex = 0 ;
  {
    do
    {
      EXEC SQL FETCH read_orderline_cur

            INTO :ol_i_id , :ol_supply_w_id , :ol_quantity , :ol_amount , :ol_delivery_d ;

      if ( sqlca.sqlcode == 0 )
      {
        ordstat->item[ itemArrayIndex ].s_OL_I_ID          = ol_i_id ;
        ordstat->item[ itemArrayIndex ].s_OL_SUPPLY_W_ID   = ol_supply_w_id ;
        ordstat->item[ itemArrayIndex ].s_OL_QUANTITY      = ol_quantity ;
        ordstat->item[ itemArrayIndex ].s_OL_AMOUNT        = ol_amount ;
        ordstat->item[ itemArrayIndex ].s_OL_DELIVERY_D_time = ol_delivery_d ;

        itemArrayIndex++;
      }
      else
      if (sqlca.sqlcode < 0 )
      {
        DLCHK( retry_tran ) ;
        sqlerror( ORDSTAT_SQL, "FETCH CURSOR read_orderline_cur" , __LINE__ , &sqlca ) ;
        goto ferror ;
      }
    }
    while ( sqlca.sqlcode == 0 ) ;
  }

  ordstat->s_ol_cnt = itemArrayIndex ;

  EXEC SQL COMMIT ;

  if ( sqlca.sqlcode == 0 )
  {
    ordstat->s_transtatus = TRAN_OK ;
  }
  else
  {
    DLCHK( retry_tran );
    sqlerror(ORDSTAT_SQL, "COMMIT", __LINE__ , &sqlca);
    goto ferror ;
  }

mexit:

  if ( sqlca.sqlcode >= 0 )
  {
    rc = SQLZ_HOLD_PROC ;
  }
  else
  {
    rc = SQLZ_DISCONNECT_PROC ;
```

```
  }

  return( rc ) ;

ferror:

  ordstat->s_transtatus = FATAL_SQLERROR ;

  EXEC SQL rollback work ;

  if ( sqlca.sqlcode != 0 )
  {
    sqlerror(ORDSTAT_SQL, "ROLLBACK FAILED", __LINE__, &sqlca);
  }

  goto mexit;
}

// --------------------------------------------------------------------
// Delivery CLIENT
// --------------------------------------------------------------------


int delivery_sql (   struct in_delivery_struct  * in_delivery
                   , struct out_delivery_struct * delivery      )
{
  struct sqlca sqlca ;

  EXEC SQL BEGIN DECLARE SECTION;

    struct vc_del_in
    {
      short len ;
      char data[ 22 ] ;
    } * in_del ;

    struct vc_del_out
    {
      short len;
      char data[ 50 ] ;
    } * out_del ;

  EXEC SQL END DECLARE SECTION;

  int rc = 0;

  /* Create Timestamp */

  in_delivery->s_O_DELIVERY_D_time = (sqlint64) time( NULL ) ;

  in_del      = (struct vc_del_in *) in_delivery ;
  in_del->len = sizeof(struct in_delivery_struct) - SPGENERAL_ADJUST;

  out_del      = (struct vc_del_out *) delivery ;
  out_del->len = sizeof(struct out_delivery_struct) - SPGENERAL_ADJUST;

#ifdef DEBUGIT
  del_debug(delivery, in_delivery, "Client before SP call");
#endif /* DEBUGIT */

  EXEC SQL CALL dels ( :*in_del, :*out_del ) ;

#ifdef DEBUGIT
  del_debug(delivery, in_delivery, "Client after SP call");
#endif /* DEBUGIT */

  if ( sqlca.sqlcode )
  {
    sqlerror( DELIVERY_SQL, "DEL", __LINE__, &sqlca) ;
    del_debug( delivery, in_delivery, "sqlcode != 0" ) ;
    delivery->s_transtatus = FATAL_SQLERROR;
    rc = -1 ;
  }

  if ( delivery->s_transtatus <= FATAL_SQLERROR )
  {
    del_debug(delivery, in_delivery, "s_transtatus <= FATAL_SQLERROR");
    rc = -1 ;
  }

  return ( rc ) ;
}


// --------------------------------------------------------------------
// Delivery SERVER
// --------------------------------------------------------------------

#undef w_id
#undef d_id
#undef o_carrier_id
#undef c_id

SQL_API_RC delivery_internal ( char * pin, char * pout )
{
  struct in_delivery_struct * in_delivery = (struct in_delivery_struct *)  pin ;
  struct out_delivery_struct * delivery   = (struct out_delivery_struct *) pout ;

  struct sqlca sqlca ;

  int rc = 0 ;

  short district_id ;
  sqlint32  customer_id ;

  EXEC SQL BEGIN DECLARE SECTION;

    // input

    //##sqlint32   w_id ;
    //##short      d_id ;
    //##sqlint32   c_id ;
    //##short      o_carrier_id ;
    //##sqlint64   ol_delivery_d ;

    // output

    short      no_o_id_indicator = 0 ;
    sqlint32   no_o_id ;

  EXEC SQL END DECLARE SECTION;

  #define d_id          district_id
  #define c_id          customer_id

  #define w_id          in_delivery->s_W_ID
  #define o_carrier_id  in_delivery->s_O_CARRIER_ID
  #define ol_delivery_d in_delivery->s_O_DELIVERY_D_time

  delivery->deadlocks = -1 ;

retry_tran:

  delivery->deadlocks++;

  for ( d_id = 1 ;  d_id <= DISTRICTS_PER_WAREHOUSE ; d_id++ )
  {
    no_o_id = 0 ;
    no_o_id_indicator = 0 ;

      EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

        SELECT O_ID

                  INTO :no_o_id :no_o_id_indicator

          FROM TABLE (  DEL( :w_id , :d_id , :o_carrier_id , :ol_delivery_d ) ) AS T ;

        COMMIT ;

      END COMPOUND ;

    if ( sqlca.sqlcode == 0 )
    {
      /* Refer to clause 2.7.4.2, bullet 3 in spec.*/
      /* Need to report if more than 1 or 1% of   */
      /* no_o_id will remain 0 if null returned, so just treat the same way */

      delivery->s_O_ID[ d_id - 1 ] = no_o_id ;
    }
    else
    {
      DLCHK( retry_tran );

      sqlerror( DELIVERY_SQL , "DELIVERY", __LINE__ , &sqlca);
      goto ferror ;
    }
  }

  delivery->s_transtatus = TRAN_OK ;

mexit:

  if ( sqlca.sqlcode >= 0 )
  {
    rc = SQLZ_HOLD_PROC ;
  }
  else
  {
    rc = SQLZ_DISCONNECT_PROC ;
  }

  return( rc ) ;

ferror:

  delivery->s_transtatus = FATAL_SQLERROR ;

  EXEC SQL rollback work ;

  if ( sqlca.sqlcode != 0 )
  {
    sqlerror( DELIVERY_SQL, "ROLLBACK FAILED", __LINE__ , &sqlca ) ;
  }

  goto mexit ;
}
```

```
// --------------------------------------------------------------------
// Stock CLIENT
// --------------------------------------------------------------------

#undef w_id
#undef d_id

int stocklev_sql (   struct in_stocklev_struct  * in_stocklev
                   , struct out_stocklev_struct * stocklev     )
{
  struct sqlca sqlca ;

  int rc = 0 ;

  EXEC SQL BEGIN DECLARE SECTION;

    // input

    //##sqlint32   w_id ;
    //##short     d_id ;
    sqlint32      threshold ;

    // output

    sqlint32    low_stock ;

  EXEC SQL END DECLARE SECTION;

  #define w_id      in_stocklev->s_W_ID
  #define d_id      in_stocklev->s_D_ID
  #define threshold in_stocklev->s_threshold
  #define low_stock stocklev->s_low_stock

  stocklev->deadlocks = -1 ;

#ifdef DEBUGIT
  stk_debug(stocklev, in_stocklev, "Client before SP call");
#endif /* DEBUGIT */

retry_tran:

  stocklev->deadlocks ++ ;

  EXEC SQL BEGIN COMPOUND NOT ATOMIC STATIC

    SELECT COUNT( S_I_ID ) INTO :low_stock

      FROM ( SELECT DISTINCT S_I_ID

               FROM ORDER_LINE , STOCK , DISTRICT

              WHERE D_W_ID = :w_id
                AND D_ID = :d_id
                AND OL_O_ID < d_next_o_id
                AND OL_O_ID >= ( d_next_o_id - 20 )
                AND OL_W_ID = D_W_ID
                AND OL_D_ID = D_ID
                AND S_I_ID = OL_I_ID
                AND S_W_ID = OL_W_ID
                AND S_QUANTITY < :threshold

           ) OLS

      WITH CS
    ;

    COMMIT ;

  END COMPOUND ;

#ifdef DEBUGIT
  stk_debug(stocklev, in_stocklev, "Client after SP call");
#endif /* DEBUGIT */

  if ( sqlca.sqlcode == 0 )
  {
    stocklev->s_transtatus = TRAN_OK ;
  }
  else
  {
    DLCHK( retry_tran ) ;

    sqlerror( STOCKLEV_SQL , "STK" , __LINE__ , &sqlca);
    goto ferror ;
  }

mexit:

  if ( sqlca.sqlcode )
  {
    sqlerror( STOCKLEV_SQL, "STK", __LINE__, &sqlca) ;
    stk_debug( stocklev, in_stocklev, "sqlcode != 0" ) ;
    stocklev->s_transtatus = FATAL_SQLERROR;
    rc = -1 ;
  }

  if ( stocklev->s_transtatus <= FATAL_SQLERROR )
  {
    stk_debug(stocklev, in_stocklev, "s_transtatus <= FATAL_SQLERROR");
    rc = -1 ;
  }

  return rc ;

ferror:

  stocklev->s_transtatus = FATAL_SQLERROR ;

  EXEC SQL ROLLBACK WORK ;

  if ( sqlca.sqlcode != 0 )
  {
    sqlerror( STOCKLEV_SQL, "ROLLBACK FAILED", __LINE__, &sqlca ) ;
  }

  goto mexit ;
}

// --------------------------------------------------------------------
// Stored Procedure Stubs
// --------------------------------------------------------------------

SQL_API_RC SQL_API_FN news( char *pin, char *pout )
{
  return new_order_internal( pin, pout ) ;
}

SQL_API_RC SQL_API_FN ords( char *pin, char *pout )
{
  return order_status_internal( pin, pout ) ;
}

SQL_API_RC SQL_API_FN dels ( char * pin, char * pout )
{
  return delivery_internal( pin, pout ) ;

}
```

tpccenv.sh

```
#############################################################################
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#############################################################################

#
# tpccenv.sh - UNIX Environment Setup
#

# The DB2 Instance Name (for DB2)
export DB2INSTANCE=${USER}

# The OS being used (i.e. "UNIX", "WINDOWS")
export PLATFORM=UNIX

# The type of make command and slash used by the OS.
# (i.e. UNIX - "/", WINDOWS - "\").
# These are referenced all over the kit.
export SLASH="/";
export MAKE=make

# Specifies whether or not to use dari stored proc's for the TPC-C driver.  Set to either DARIVERSION or
NONDARI;
#export TPCC_SPTYPE=NOSP
#export TPCC_SPTYPE=SPGENERAL2
export TPCC_SPTYPE=SPGENERAL
#export TPCC_SPTYPE=DARI2SQLDA

export DB2VERSION=v8

# The schema name is typically the SQL autorization ID (or username).
# This is required for runstats and EEE.
export TPCC_SCHEMA=${USER}

# DB2 EE/EEE Configuration
export DB2EDITION=EE
#export DB2EDITION=EEE
export DB2NODE=0
export DB2NODES=1;              # set to the number of nodes you have.  Set to 1 for EE.

# TPCC General Configuration
export TPCC_DBNAME=TPCC
export TPCC_ROOT=${HOME}/tpc-c.ibm
export TPCC_SQLLIB=${HOME}/sqllib
export TPCC_RUNDATA=${HOME}/tpccdata

# TPCC Debug Configuration
```

```
# This is the path where all error and debug logs are placed.
export TPCC_DEBUGDIR=/tmp

# Specifies where stored procedures should be placed and if they should
# be fenced.
export TPCC_SPDIR=${TPCC_SQLLIB}/function
export TPCC_FENCED=NO
```

tpccdbg.h

```
/*************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*************************************************************************/

/*
 * tpccdbg.h - Debugging Macros
 */

#ifndef __TPCCDBG_H
#define __TPCCDBG_H


extern void new_debug (struct out_neword_struct *neword_ptr,
                       struct in_neword_struct *in_neword_ptr,
                       char *msg);
extern void pay_debug (struct out_payment_struct *payment_ptr,
                       struct in_payment_struct *in_payment_ptr,
                       char *msg);
extern void ord_debug (struct out_ordstat_struct *ordstat_ptr,
                       struct in_ordstat_struct *in_ordstat_ptr,
                       char *msg);
extern void del_debug (struct out_delivery_struct *delivery_ptr,
                       struct in_delivery_struct *in_delivery_ptr,
                       char *msg);
extern void stk_debug (struct out_stocklev_struct *stocklev_ptr,
                       struct in_stocklev_struct *in_stocklev_ptr,
                       char *msg);

extern void new_print (struct out_neword_struct *neword_ptr,
                       struct in_neword_struct *in_neword_ptr,
                       char *filename,
                       char *msg);
extern void pay_print (struct out_payment_struct *payment_ptr,
                       struct in_payment_struct *in_payment_ptr,
                       char *filename,
                       char *msg);
extern void ord_print (struct out_ordstat_struct *ordstat_ptr,
                       struct in_ordstat_struct *in_ordstat_ptr,
                       char *filename,
                       char *msg);
extern void del_print (struct out_delivery_struct *delivery_ptr,
                       struct in_delivery_struct *in_delivery_ptr,
                       char *filename,
                       char *msg);
extern void stk_print (struct out_stocklev_struct *stocklev_ptr,
                       struct in_stocklev_struct *in_stocklev_ptr,
                       char *filename,
                       char *msg);


#endif // __TPCCDBG_
```

tpccapp.h

```
/*************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*************************************************************************/

/*
 * tpccapp.h - Application Macros
 */

#ifndef __TPCCAPP_H
#define __TPCCAPP_H

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <time.h>

#define daricall

#include "sqlca.h"
#include "sqlcodes.h"


/*****************************************************************************/
/* SQLCODE Macros                                                            */
/*****************************************************************************/

#define DLCHK(a) \
  if (sqlca.sqlcode == SQL_RC_E911) { goto a; }

#define NACOMPCHK(last) \
  if (sqlca.sqlcode != SQL_RC_E1339) { last = -1; } \
  else { int a = ((sqlca.sqlerrmc[4] == 0x20) ? 0 : sqlca.sqlerrmc[4]-0x30); \
         int b = ((sqlca.sqlerrmc[5] == 0x20) ? 0 : sqlca.sqlerrmc[5]-0x30); \
         if (b == 0) { last = a; } else { last = a * 10 + b; } \
       }


#endif // __TPCCAPP_H
```

tpccctx.sqc

```
/*************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*************************************************************************/

/*
 * tpccctx.sqc - TPCC context code
 */

#include <stdlib.h>
#include <stdio.h>
#include <sqlutil.h>

int connect_to_TM(char *in_dbname);
int disconnect_from_TM(void);

int connect_to_TM(char *in_dbname)
{
  SQL_STRUCTURE sqlca  sqlca;
  void              *ctx;

  EXEC SQL BEGIN DECLARE SECTION;
    char dbname[9];
  EXEC SQL END DECLARE SECTION;


  strncpy(dbname,in_dbname,8);

  EXEC SQL CONNECT TO :dbname IN SHARE MODE;

  if (SQLCODE != 0) { return SQLCODE; }

  return 0;

} /* connect_to_TM */

int disconnect_from_TM(void)
{
  SQL_STRUCTURE sqlca  sqlca;
  void              *ctx;

  EXEC SQL CONNECT RESET;

  if (SQLCODE != 0) { return SQLCODE; }


  return 0;

} /* disconnect_from_TM */
```

uncat-func.ddl

```
-------------------------------------------------------------------------
-- Licensed Materials - Property of IBM
```

```
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
--------------------------------------------------------------------------------
--
-- uncat-func.ddl - Drop table function DDL
--

-- STOCK LEVEL
DROP SPECIFIC FUNCTION STOCK_LEVEL $
-- DELIVERY
DROP SPECIFIC FUNCTION DELIVERY $
-- ORDER STATUS
DROP SPECIFIC FUNCTION ORD_C_LAST $
DROP SPECIFIC FUNCTION ORD_C_ID $
-- PAYMENT
DROP SPECIFIC FUNCTION PAY_C_LAST $
DROP SPECIFIC FUNCTION PAY_C_ID $
-- NEW ORDER
DROP SPECIFIC FUNCTION NEW_OL_ALL $
DROP SPECIFIC FUNCTION NEW_OL_LOCAL $
DROP SPECIFIC FUNCTION NEW_WH $
```

uncat-proc.ddl

```
DROP PROCEDURE news
                (varchar(270),varchar(662));
DROP PROCEDURE news
                (varchar(214),varchar(790));
DROP PROCEDURE news;


DROP PROCEDURE pays
                (varchar(62),varchar(534));
DROP PROCEDURE pays;

DROP PROCEDURE ords
                (varchar(42),varchar(446));
DROP PROCEDURE ords
                (varchar(34),varchar(574));
DROP PROCEDURE ords;

DROP PROCEDURE dels
                (varchar(22),varchar(50));
DROP PROCEDURE dels
                (varchar(22),varchar(50));
DROP PROCEDURE dels;

DROP PROCEDURE stks
                (varchar(18),varchar(14));
DROP PROCEDURE stks
                (varchar(14),varchar(14));
DROP PROCEDURE stks;
```

Appendix B: Tunable Parameters

B.1 Database Parameters

```
#####################################################################
#db2 get db cfg for tpcc


     Database Configuration for Database TPCC

Database configuration release level            = 0x0a00
Database release level                          = 0x0a00

Database territory                              = US
Database code page                              =  819
Database code set                               = ISO8859-1
Database country/region code                    = 1

Dynamic SQL Query management         (DYN_QUERY_MGMT) = DISABLE

Discovery support for this database      (DISCOVER_DB) = ENABLE

Default query optimization class       (DFT_QUERYOPT) = 5
Degree of parallelism                  (DFT_DEGREE) = 1
Continue upon arithmetic exceptions    (DFT_SQLMATHWARN) = NO
Default refresh age                    (DFT_REFRESH_AGE) = 0
Number of frequent values retained     (NUM_FREQVALUES) = 10
Number of quantiles retained           (NUM_QUANTILES) = 20

Backup pending                                  = NO

Database is consistent                          = NO
Rollforward pending                             = NO
Restore pending                                 = NO

Multi-page file allocation enabled              = NO

Log retain for recovery status                  = RECOVERY
User exit for logging status                    = NO

Data Links Token Expiry Interval (sec)  (DL_EXPINT) = 60
Data Links Write Token Init Expiry Intvl(DL_WT_IEXPINT) = 60
Data Links Number of Copies             (DL_NUM_COPIES) = 1
Data Links Time after Drop (days)       (DL_TIME_DROP) = 1
Data Links Token in Uppercase           (DL_UPPER) = NO
Data Links Token Algorithm              (DL_TOKEN) = MAC0

Database heap (4KB)                          (DBHEAP) = 40000
Size of database shared memory (4KB)  (DATABASE_MEMORY) = AUTOMATIC
Catalog cache size (4KB)              (CATALOGCACHE_SZ) = (MAXAPPLS*4)
Log buffer size (4KB)                    (LOGBUFSZ) = 8400
Utilities heap size (4KB)              (UTIL_HEAP_SZ) = 5000
Buffer pool size (pages)                  (BUFFPAGE) = 1000
Extended storage segments size (4KB)    (ESTORE_SEG_SZ) = 16000
Number of extended storage segments   (NUM_ESTORE_SEGS) = 0
Max storage for lock list (4KB)           (LOCKLIST) = 8500

Max size of appl. group mem set (4KB) (APPGROUP_MEM_SZ) = 20000
Percent of mem for appl. group heap   (GROUPHEAP_RATIO) = 70
Max appl. control heap size (4KB)     (APP_CTL_HEAP_SZ) = 128

Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = (SHEAPTHRES)
Sort list heap (4KB)                     (SORTHEAP) = 16
SQL statement heap (4KB)                 (STMTHEAP) = 55000
Default application heap (4KB)           (APPLHEAPSZ) = 2500
Package cache size (4KB)                 (PCKCACHESZ) = 50000
Statistics heap size (4KB)               (STAT_HEAP_SZ) = 4384

Interval for checking deadlock (ms)        (DLCHKTIME) = 3000
Percent. of lock lists per application     (MAXLOCKS) = 20
Lock timeout (sec)                      (LOCKTIMEOUT) = -1

Changed pages threshold                 (CHNGPGS_THRESH) = 99
Number of asynchronous page cleaners   (NUM_IOCLEANERS) = 1
Number of I/O servers                   (NUM_IOSERVERS) = 1
Index sort flag                         (INDEXSORT) = YES
Sequential detect flag                  (SEQDETECT) = NO
Default prefetch size (pages)         (DFT_PREFETCH_SZ) = 32

Track modified pages                       (TRACKMOD) = OFF

Default number of containers                    = 1
Default tablespace extentsize (pages)   (DFT_EXTENT_SZ) = 32

Max number of active applications          (MAXAPPLS) = 2000
Average number of active applications      (AVG_APPLS) = 1000
Max DB files open per application          (MAXFILOP) = 800

Log file size (4KB)                     (LOGFILSIZ) = 65535
Number of primary log files               (LOGPRIMARY) = 250
Number of secondary log files             (LOGSECOND) = 0
Changed path to log files                 (NEWLOGPATH) =
Path to log files                               = /dev/rdb2loglv
Overflow log path                      (OVERFLOWLOGPATH) =
Mirror log path                        (MIRRORLOGPATH) =
First active log file                           = S0000148.LOG
Block log on disk full                 (BLK_LOG_DSK_FUL) = NO
Percent of max active log space by transaction(MAX_LOG) = 0
Num. of active log files for 1 active UOW(NUM_LOG_SPAN) = 0

Group commit count                        (MINCOMMIT) = 3
Percent log file reclaimed before soft chckpt (SOFTMAX) = 19500
Log retain for recovery enabled           (LOGRETAIN) = RECOVERY
User exit for logging enabled             (USEREXIT) = OFF

Auto restart enabled                    (AUTORESTART) = ON
Index re-creation time                    (INDEXREC) = SYSTEM (RESTART)
Default number of loadrec sessions    (DFT_LOADREC_SES) = 1
Number of database backups to retain   (NUM_DB_BACKUPS) = 12
Recovery history retention (days)     (REC_HIS_RETENTN) = 366

TSM management class                  (TSM_MGMTCLASS) =
TSM node name                         (TSM_NODENAME) =
TSM owner                             (TSM_OWNER) =
TSM password                          (TSM_PASSWORD) =


#####################################################################
#db2 get dbm cfg


        Database Manager Configuration

   Node type = Database Server with local clients

Database manager configuration release level            = 0x0a00

CPU speed (millisec/instruction)             (CPUSPEED) = 3.975564e-07

Max number of concurrently active databases     (NUMDB) = 1
Data Links support                          (DATALINKS) = NO
Federated Database System Support           (FEDERATED) = NO
Transaction processor monitor name        (TP_MON_NAME) =

Default charge-back account          (DFT_ACCOUNT_STR) =

Java Development Kit installation path       (JDK_PATH) = /usr/java13_64

Diagnostic error capture level              (DIAGLEVEL) = 1
Notify Level                            (NOTIFYLEVEL) = 3
Diagnostic data directory path              (DIAGPATH) =

Default database monitor switches
  Buffer pool                       (DFT_MON_BUFPOOL) = OFF
  Lock                              (DFT_MON_LOCK) = OFF
  Sort                              (DFT_MON_SORT) = OFF
  Statement                         (DFT_MON_STMT) = OFF
  Table                             (DFT_MON_TABLE) = OFF
  Timestamp                         (DFT_MON_TIMESTAMP) = OFF
  Unit of work                      (DFT_MON_UOW) = OFF
Monitor health of instance and databases   (HEALTH_MON) = OFF

SYSADM group name                      (SYSADM_GROUP) = STAFF
SYSCTRL group name                     (SYSCTRL_GROUP) =
SYSMAINT group name                    (SYSMAINT_GROUP) =
SYSMON group name                      (SYSMON_GROUP) =

Database manager authentication        (AUTHENTICATION) = CLIENT
Cataloging allowed without authority   (CATALOG_NOAUTH) = YES
Trust all clients                      (TRUST_ALLCLNTS) = YES
Trusted client authentication          (TRUST_CLNTAUTH) = CLIENT
Bypass federated authentication            (FED_NOAUTH) = NO

Default database path                     (DFTDBPATH) = /home/tpcc

Database monitor heap size (4KB)          (MON_HEAP_SZ) = 4096
Java Virtual Machine heap size (4KB)      (JAVA_HEAP_SZ) = 1024
Audit buffer size (4KB)                   (AUDIT_BUF_SZ) = 0
Size of instance shared memory (4KB)  (INSTANCE_MEMORY) = AUTOMATIC
Backup buffer default size (4KB)          (BACKBUFSZ) = 1024
Restore buffer default size (4KB)         (RESTBUFSZ) = 1024

Sort heap threshold (4KB)                 (SHEAPTHRES) = 20000

Directory cache support                   (DIR_CACHE) = YES

Application support layer heap size (4KB)   (ASLHEAPSZ) = 15
Max requester I/O block size (bytes)        (RQRIOBLK) = 4096
Query heap size (4KB)                     (QUERY_HEAP_SZ) = 1000
DRDA services heap size (4KB)             (DRDA_HEAP_SZ) = 128

Workload impact by throttled utilities(UTIL_IMPACT_LIM) = 100

Priority of agents                        (AGENTPRI) = 60
Max number of existing agents             (MAXAGENTS) = 2000
Agent pool size                       (NUM_POOLAGENTS) = 1820
Initial number of agents in pool       (NUM_INITAGENTS) = 1820
Max number of coordinating agents     (MAX_COORDAGENTS) = MAXAGENTS
Max no. of concurrent coordinating agents  (MAXCAGENTS) = MAX_COORDAGENTS
Max number of client connections      (MAX_CONNECTIONS) = MAX_COORDAGENTS

Keep fenced process                       (KEEPFENCED) = YES
```

```
Number of pooled fenced processes        (FENCED_POOL) = MAX_COORDAGENTS
Initialize fenced process with JVM     (INITFENCED_JVM) = NO
Initial number of fenced processes     (NUM_INITFENCED) = 0

Index re-creation time                     (INDEXREC) = RESTART

Transaction manager database name        (TM_DATABASE) = 1ST_CONN
Transaction resync interval (sec)     (RESYNC_INTERVAL) = 180

SPM name                                 (SPM_NAME) =
SPM log size                        (SPM_LOG_FILE_SZ) = 256
SPM resync agent limit                (SPM_MAX_RESYNC) = 20
SPM log path                            (SPM_LOG_PATH) =

TCP/IP Service name                       (SVCENAME) = db2ctpcc
Discovery mode                          (DISCOVER) = SEARCH
Discover server instance                (DISCOVER_INST) = ENABLE

Maximum query degree of parallelism   (MAX_QUERYDEGREE) = ANY
Enable intra-partition parallelism     (INTRA_PARALLEL) = NO

No. of int. communication buffers(4KB)(FCM_NUM_BUFFERS) = 512



######################################################################
#db2set


[i] DB2_LGPAGE_BP=yes
[i] DB2_TRUSTED_BINDIN=ON
[i] DB2_KEEPTABLELOCK=ON
[i] DB2_EVENT_LOG_CONFIG=OFF
[i] DB2_NO_FORK_CHECK=ON
[i] DB2_APM_PERFORMANCE=ALL
[i] DB2_SELECTIVITY=YES
[i] DB2ASSUMEUPDATE=ON
[i] DB2_HASH_JOIN=OFF
[i]
DB2AFFINITIES=ALL,RSET,sys/node.03.00000,sys/node.03.00001,sys/node.03.00002,sys/node.03.00003
[i] DB2CHKSQLDA=OFF
[i] DB2COMM=tcpip
```

B.2 Transaction Monitor Parameters

```
#ident   "@(#) /home/tpcc/db2kit/db2inf/tpccdef  $Revision: 1.3 $"

*RESOURCES
IPCKEY              32769

DOMAINID  tpcckit
MASTER              tpcc
MAXACCESSERS        1000
MAXSERVERS          200
MAXSERVICES         200
MAXBUFSTYPE         200
MODEL               SHM
LDBAL               Y
OPTIONS             NO_AA,NO_XA

*MACHINES
DEFAULT:
APPDIR="/home/tpcc/db2kit/db2inf"
TUXCONFIG="/home/tpcc/db2kit/db2inf/tpcccfg"
TUXDIR="/usr/local/bea/tuxedo8.0"

<HOSTNAME>          LMID=tpcc

*GROUPS
GROUP1
        LMID=tpcc GRPNO=1   OPENINFO=NONE
GROUP2
        LMID=tpcc GRPNO=2   OPENINFO=NONE
GROUP3
        LMID=tpcc GRPNO=3   OPENINFO=NONE
GROUP4
        LMID=tpcc GRPNO=4   OPENINFO=NONE
GROUP5
        LMID=tpcc GRPNO=5   OPENINFO=NONE
GROUP6
        LMID=tpcc GRPNO=6   OPENINFO=NONE

*SERVERS
DEFAULT:
                    CLOPT="-A" REPLYQ=Y

tuxServer SRVGRP=GROUP1 SRVID=1001 MIN=26 MAX=28 CLOPT="-s
ORDER:tux_new_order" RQADDR=newo
tuxServer SRVGRP=GROUP2 SRVID=2001 MIN=22          MAX=24 CLOPT="-s
PAYMENT:tux_payment" RQADDR=pmt
tuxServer SRVGRP=GROUP3 SRVID=3001 MIN=3           MAX=8 CLOPT="-s
STATUS:tux_order_status" RQADDR=ostat
tuxServer SRVGRP=GROUP4 SRVID=4001 MIN=12          MAX=13 CLOPT="-s
DELIVERY:tux_delivery -- -d" RQADDR=dlvry
tuxServer SRVGRP=GROUP5 SRVID=5001 MIN=3           MAX=8 CLOPT="-s
STOCK:tux_stock_level" RQADDR=stklvl
tuxServer SRVGRP=GROUP6 SRVID=6001 MIN=1           MAX=2  CLOPT="-s
INFO:tux_db_info -- -x"

*SERVICES
INFO
ORDER
PAYMENT
STATUS
DELIVERY
STOCK
```

B.3 AIX Parameters

IBM eServer pSeries 690

lsattr -EL

```
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     128          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      4194304      Amount of usable physical memory in Kbytes       False
conslogin    enable       System Console Login                             False
fwversion    IBM,M2P021208 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       6            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,7025-6F1 Machine name                                     False
systemid     IBM,01109F38A 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    133796000    System Bus Frequency                             False
```

lsattr -El aio0 -a maxreqs

```
maxreqs 60000 Maximum number of REQUESTS True
```

vmo -L

```
NAME                  CUR    DEF    BOOT   MIN    MAX    UNIT          TYPE
  DEPENDENCIES
--------------------------------------------------------------------------------
memory_frames         128M   128M          4KB pages      S
--------------------------------------------------------------------------------
pinnable_frames      5241K  5241K          4KB pages      S
--------------------------------------------------------------------------------
maxfree               1K     128    1K     16     200K   4KB pages      D
  minfree
  memory_frames
--------------------------------------------------------------------------------
minfree             512    1016   512    8      200K   4KB pages      D
  maxfree
  memory_frames
--------------------------------------------------------------------------------
minperm%             20     20     20     1      100    % memory       D
  maxperm%
--------------------------------------------------------------------------------
minperm            2500K  2500K                        S
--------------------------------------------------------------------------------
maxperm%             80     80     80     1      100    % memory       D
  minperm%
  maxclient%
--------------------------------------------------------------------------------
maxperm           10001K 10001K                        S
--------------------------------------------------------------------------------
strict_maxperm       0      0      0      0      1      boolean        D
--------------------------------------------------------------------------------
maxpin%             97     92     97     1      99     % memory       D
  pinnable_frames
  memory_frames
--------------------------------------------------------------------------------
maxpin            127139K 127139K                      S
--------------------------------------------------------------------------------
maxclient%           80     80     80     1      100    % memory       D
  maxperm%
--------------------------------------------------------------------------------
lrubucket           128K   128K   128K   64K           4KB pages      D
--------------------------------------------------------------------------------
defps                1      1      1      0      1      boolean        D
--------------------------------------------------------------------------------
nokilluid            0      0      0      0      2047M  uid            D
--------------------------------------------------------------------------------
numpsblks         1680K  1680K          4KB pages      S
--------------------------------------------------------------------------------
npskill           13440  13440  13440  1      1679K  4KB pages      D
--------------------------------------------------------------------------------
npswarn           53760  53760  53760  0      1679K  4KB pages      D
--------------------------------------------------------------------------------
v_pinshm             1      0      1      0      1      boolean        D
--------------------------------------------------------------------------------
pta_balance_threshold  n/a   50     50     1      99     % pta segment  R
--------------------------------------------------------------------------------
pagecoloring         n/a    0      0      0      1      boolean        B
--------------------------------------------------------------------------------
framesets            2      2      2      1      10             B
--------------------------------------------------------------------------------
mempools             1      1      1      1      32             B
--------------------------------------------------------------------------------
lgpg_size           16M    0      16M    0      256M   bytes          B
  lgpg_regions
--------------------------------------------------------------------------------
lgpg_regions        28000  0      28000  0                     B
  lgpg_size
--------------------------------------------------------------------------------
num_spec_dataseg     0      0      0      0                     B
--------------------------------------------------------------------------------
spec_dataseg_int     512    512    512    0                     B
--------------------------------------------------------------------------------
memory_affinity      1      1      1      0      1      boolean        B
--------------------------------------------------------------------------------
```

n/a means parameter not supported by the current platform or kernel

Parameter types:
  S = Static: cannot be changed
  D = Dynamic: can be freely changed
  B = Bosboot: can only be changed using bosboot and reboot
  R = Reboot: can only be changed during reboot
  C = Connect: changes are only effective for future socket connections
  M = Mount: changes are only effective for future mountings
  I = Incremental: can only be incremented

Value conventions:
  K = Kilo: 2^10   G = Giga: 2^30   P = Peta: 2^50
  M = Mega: 2^20   T = Tera: 2^40   E = Exa: 2^60

ioo -L

```
NAME                  CUR    DEF    BOOT   MIN    MAX    UNIT          TYPE
  DEPENDENCIES
--------------------------------------------------------------------------------
minpgahead           2      2      2      0      4K     4KB pages      D
  maxpgahead
```

```
------------------------------------------------------------------------------
maxpgahead              8     8     8     0     4K    4KB pages        D
   minpgahead
------------------------------------------------------------------------------
pd_npages              64K   64K   64K   1     512K  4KB pages        D
------------------------------------------------------------------------------
maxrandwrt              0     0     0     0     512K  4KB pages        D
------------------------------------------------------------------------------
numclust                1     1     1     0     2047M 16KB/cluster     D
------------------------------------------------------------------------------
numfsbufs              1000  196   1000  1     2047M                  M
------------------------------------------------------------------------------
sync_release_ilock      0     0     0     0     1     boolean          D
------------------------------------------------------------------------------
lvm_bufcnt              9     9     9     1     64    128KB/buffer     D
------------------------------------------------------------------------------
j2_minPageReadAhead     2     2     2     0     128   4KB pages        D
------------------------------------------------------------------------------
j2_maxPageReadAhead     8     8     8     0     128   4KB pages        D
------------------------------------------------------------------------------
j2_nBufferPerPagerDevice 512   512   512   0     2047M                  M
------------------------------------------------------------------------------
j2_nPagesPerWriteBehindCluster
                       32    32    32    0     128                    D
------------------------------------------------------------------------------
j2_maxRandomWrite       0     0     0     0     128   4KB pages        D
------------------------------------------------------------------------------
j2_nRandomCluster       0     0     0     0     2047M 16KB clusters    D
------------------------------------------------------------------------------
hd_pvs_opn             244         244                                S
------------------------------------------------------------------------------
hd_pbuf_cnt            48510 31K   16894 0     2047M                  I
------------------------------------------------------------------------------

n/a means parameter not supported by the current platform or kernel

Parameter types:
  S = Static: cannot be changed
  D = Dynamic: can be freely changed
  B = Bosboot: can only be changed using bosboot and reboot
  R = Reboot: can only be changed during reboot
  C = Connect: changes are only effective for future socket connections
  M = Mount: changes are only effective for future mountings
  I = Incremental: can only be incremented
```

Appendix C: Database Setup Code

C.1 Database Creation Scripts

ALTTBSP_PF_0.ddl

```
connect to TPCC;
alter tablespace is_customer_01 prefetchsize 0;
alter tablespace is_customer_02 prefetchsize 0;
alter tablespace is_customer_03 prefetchsize 0;
alter tablespace is_customer_04 prefetchsize 0;
alter tablespace is_customer_05 prefetchsize 0;
alter tablespace is_customer_06 prefetchsize 0;
alter tablespace is_customer_07 prefetchsize 0;
alter tablespace is_customer_08 prefetchsize 0;
alter tablespace is_customer_09 prefetchsize 0;
alter tablespace is_customer_10 prefetchsize 0;
alter tablespace is_customer_11 prefetchsize 0;
alter tablespace is_customer_12 prefetchsize 0;
alter tablespace is_customer_13 prefetchsize 0;
alter tablespace is_customer_14 prefetchsize 0;
alter tablespace is_customer_15 prefetchsize 0;
alter tablespace is_customer_16 prefetchsize 0;
alter tablespace is_customer_17 prefetchsize 0;
alter tablespace is_customer_18 prefetchsize 0;
alter tablespace is_customer_19 prefetchsize 0;
alter tablespace is_customer_20 prefetchsize 0;
alter tablespace is_customer_21 prefetchsize 0;
alter tablespace is_customer_22 prefetchsize 0;
alter tablespace is_customer_23 prefetchsize 0;
alter tablespace is_customer_24 prefetchsize 0;
alter tablespace is_customer_25 prefetchsize 0;
alter tablespace is_customer_26 prefetchsize 0;
alter tablespace is_customer_27 prefetchsize 0;
alter tablespace is_customer_28 prefetchsize 0;
alter tablespace is_neword_01 prefetchsize 0;
alter tablespace is_order_01 prefetchsize 0;
alter tablespace ts_customer_01 prefetchsize 0;
alter tablespace ts_customer_02 prefetchsize 0;
alter tablespace ts_customer_03 prefetchsize 0;
alter tablespace ts_customer_04 prefetchsize 0;
alter tablespace ts_customer_05 prefetchsize 0;
alter tablespace ts_customer_06 prefetchsize 0;
alter tablespace ts_customer_07 prefetchsize 0;
alter tablespace ts_customer_08 prefetchsize 0;
alter tablespace ts_customer_09 prefetchsize 0;
alter tablespace ts_customer_10 prefetchsize 0;
alter tablespace ts_customer_11 prefetchsize 0;
alter tablespace ts_customer_12 prefetchsize 0;
alter tablespace ts_customer_13 prefetchsize 0;
alter tablespace ts_customer_14 prefetchsize 0;
alter tablespace ts_customer_15 prefetchsize 0;
alter tablespace ts_customer_16 prefetchsize 0;
alter tablespace ts_customer_17 prefetchsize 0;
alter tablespace ts_customer_18 prefetchsize 0;
alter tablespace ts_customer_19 prefetchsize 0;
alter tablespace ts_customer_20 prefetchsize 0;
alter tablespace ts_customer_21 prefetchsize 0;
alter tablespace ts_customer_22 prefetchsize 0;
alter tablespace ts_customer_23 prefetchsize 0;
alter tablespace ts_customer_24 prefetchsize 0;
alter tablespace ts_customer_25 prefetchsize 0;
alter tablespace ts_customer_26 prefetchsize 0;
alter tablespace ts_customer_27 prefetchsize 0;
alter tablespace ts_customer_28 prefetchsize 0;
alter tablespace ts_dist_01 prefetchsize 0;
alter tablespace ts_history_01 prefetchsize 0;
alter tablespace ts_item_01 prefetchsize 0;
alter tablespace ts_neword_01 prefetchsize 0;
alter tablespace ts_order_01 prefetchsize 0;
alter tablespace ts_orderline_01 prefetchsize 0;
alter tablespace ts_orderline_02 prefetchsize 0;
alter tablespace ts_orderline_03 prefetchsize 0;
alter tablespace ts_orderline_04 prefetchsize 0;
alter tablespace ts_orderline_05 prefetchsize 0;
alter tablespace ts_orderline_06 prefetchsize 0;
alter tablespace ts_orderline_07 prefetchsize 0;
alter tablespace ts_orderline_08 prefetchsize 0;
alter tablespace ts_orderline_09 prefetchsize 0;
alter tablespace ts_orderline_10 prefetchsize 0;
alter tablespace ts_orderline_11 prefetchsize 0;
alter tablespace ts_orderline_12 prefetchsize 0;
alter tablespace ts_orderline_13 prefetchsize 0;
alter tablespace ts_orderline_14 prefetchsize 0;
alter tablespace ts_orderline_15 prefetchsize 0;
alter tablespace ts_orderline_16 prefetchsize 0;
alter tablespace ts_orderline_17 prefetchsize 0;
alter tablespace ts_orderline_18 prefetchsize 0;
alter tablespace ts_orderline_19 prefetchsize 0;
alter tablespace ts_orderline_20 prefetchsize 0;
alter tablespace ts_orderline_21 prefetchsize 0;
alter tablespace ts_orderline_22 prefetchsize 0;
alter tablespace ts_orderline_23 prefetchsize 0;
alter tablespace ts_orderline_24 prefetchsize 0;
alter tablespace ts_orderline_25 prefetchsize 0;
alter tablespace ts_orderline_26 prefetchsize 0;
alter tablespace ts_orderline_27 prefetchsize 0;
alter tablespace ts_orderline_28 prefetchsize 0;
alter tablespace ts_stock_01 prefetchsize 0;
alter tablespace ts_stock_02 prefetchsize 0;
alter tablespace ts_stock_03 prefetchsize 0;
alter tablespace ts_stock_04 prefetchsize 0;
alter tablespace ts_stock_05 prefetchsize 0;
alter tablespace ts_stock_06 prefetchsize 0;
alter tablespace ts_stock_07 prefetchsize 0;
alter tablespace ts_stock_08 prefetchsize 0;
alter tablespace ts_stock_09 prefetchsize 0;
alter tablespace ts_stock_10 prefetchsize 0;
alter tablespace ts_stock_11 prefetchsize 0;
alter tablespace ts_stock_12 prefetchsize 0;
alter tablespace ts_stock_13 prefetchsize 0;
alter tablespace ts_stock_14 prefetchsize 0;
alter tablespace ts_stock_15 prefetchsize 0;
alter tablespace ts_stock_16 prefetchsize 0;
alter tablespace ts_stock_17 prefetchsize 0;
alter tablespace ts_stock_18 prefetchsize 0;
alter tablespace ts_stock_19 prefetchsize 0;
alter tablespace ts_stock_20 prefetchsize 0;
alter tablespace ts_stock_21 prefetchsize 0;
alter tablespace ts_stock_22 prefetchsize 0;
alter tablespace ts_stock_23 prefetchsize 0;
alter tablespace ts_stock_24 prefetchsize 0;
alter tablespace ts_stock_25 prefetchsize 0;
alter tablespace ts_stock_26 prefetchsize 0;
alter tablespace ts_stock_27 prefetchsize 0;
alter tablespace ts_stock_28 prefetchsize 0;
alter tablespace ts_ware_01 prefetchsize 0;
connect reset;
```

ALTTBSP_FP_512.ddl

```
connect to TPCC;
alter tablespace is_customer_01 prefetchsize 512;
alter tablespace is_customer_02 prefetchsize 512;
alter tablespace is_customer_03 prefetchsize 512;
alter tablespace is_customer_04 prefetchsize 512;
alter tablespace is_customer_05 prefetchsize 512;
alter tablespace is_customer_06 prefetchsize 512;
alter tablespace is_customer_07 prefetchsize 512;
alter tablespace is_customer_08 prefetchsize 512;
alter tablespace is_customer_09 prefetchsize 512;
alter tablespace is_customer_10 prefetchsize 512;
alter tablespace is_customer_11 prefetchsize 512;
alter tablespace is_customer_12 prefetchsize 512;
alter tablespace is_customer_13 prefetchsize 512;
alter tablespace is_customer_14 prefetchsize 512;
alter tablespace is_customer_15 prefetchsize 512;
alter tablespace is_customer_16 prefetchsize 512;
alter tablespace is_customer_17 prefetchsize 512;
alter tablespace is_customer_18 prefetchsize 512;
alter tablespace is_customer_19 prefetchsize 512;
alter tablespace is_customer_20 prefetchsize 512;
alter tablespace is_customer_21 prefetchsize 512;
alter tablespace is_customer_22 prefetchsize 512;
alter tablespace is_customer_23 prefetchsize 512;
alter tablespace is_customer_24 prefetchsize 512;
alter tablespace is_customer_25 prefetchsize 512;
alter tablespace is_customer_26 prefetchsize 512;
alter tablespace is_customer_27 prefetchsize 512;
alter tablespace is_customer_28 prefetchsize 512;
alter tablespace is_neword_01 prefetchsize 512;
alter tablespace is_order_01 prefetchsize 512;
alter tablespace ts_customer_01 prefetchsize 512;
alter tablespace ts_customer_02 prefetchsize 512;
alter tablespace ts_customer_03 prefetchsize 512;
alter tablespace ts_customer_04 prefetchsize 512;
alter tablespace ts_customer_05 prefetchsize 512;
alter tablespace ts_customer_06 prefetchsize 512;
alter tablespace ts_customer_07 prefetchsize 512;
alter tablespace ts_customer_08 prefetchsize 512;
alter tablespace ts_customer_09 prefetchsize 512;
alter tablespace ts_customer_10 prefetchsize 512;
alter tablespace ts_customer_11 prefetchsize 512;
alter tablespace ts_customer_12 prefetchsize 512;
alter tablespace ts_customer_13 prefetchsize 512;
alter tablespace ts_customer_14 prefetchsize 512;
alter tablespace ts_customer_15 prefetchsize 512;
alter tablespace ts_customer_16 prefetchsize 512;
```

```
alter tablespace ts_customer_17 prefetchsize 512;
alter tablespace ts_customer_18 prefetchsize 512;
alter tablespace ts_customer_19 prefetchsize 512;
alter tablespace ts_customer_20 prefetchsize 512;
alter tablespace ts_customer_21 prefetchsize 512;
alter tablespace ts_customer_22 prefetchsize 512;
alter tablespace ts_customer_23 prefetchsize 512;
alter tablespace ts_customer_24 prefetchsize 512;
alter tablespace ts_customer_25 prefetchsize 512;
alter tablespace ts_customer_26 prefetchsize 512;
alter tablespace ts_customer_27 prefetchsize 512;
alter tablespace ts_customer_28 prefetchsize 512;
alter tablespace ts_dist_01 prefetchsize 512;
alter tablespace ts_history_01 prefetchsize 512;
alter tablespace ts_item_01 prefetchsize 512;
alter tablespace ts_neword_01 prefetchsize 512;
alter tablespace ts_order_01 prefetchsize 512;
alter tablespace ts_orderline_01 prefetchsize 512;
alter tablespace ts_orderline_02 prefetchsize 512;
alter tablespace ts_orderline_03 prefetchsize 512;
alter tablespace ts_orderline_04 prefetchsize 512;
alter tablespace ts_orderline_05 prefetchsize 512;
alter tablespace ts_orderline_06 prefetchsize 512;
alter tablespace ts_orderline_07 prefetchsize 512;
alter tablespace ts_orderline_08 prefetchsize 512;
alter tablespace ts_orderline_09 prefetchsize 512;
alter tablespace ts_orderline_10 prefetchsize 512;
alter tablespace ts_orderline_11 prefetchsize 512;
alter tablespace ts_orderline_12 prefetchsize 512;
alter tablespace ts_orderline_13 prefetchsize 512;
alter tablespace ts_orderline_14 prefetchsize 512;
alter tablespace ts_orderline_15 prefetchsize 512;
alter tablespace ts_orderline_16 prefetchsize 512;
alter tablespace ts_orderline_17 prefetchsize 512;
alter tablespace ts_orderline_18 prefetchsize 512;
alter tablespace ts_orderline_19 prefetchsize 512;
alter tablespace ts_orderline_20 prefetchsize 512;
alter tablespace ts_orderline_21 prefetchsize 512;
alter tablespace ts_orderline_22 prefetchsize 512;
alter tablespace ts_orderline_23 prefetchsize 512;
alter tablespace ts_orderline_24 prefetchsize 512;
alter tablespace ts_orderline_25 prefetchsize 512;
alter tablespace ts_orderline_26 prefetchsize 512;
alter tablespace ts_orderline_27 prefetchsize 512;
alter tablespace ts_orderline_28 prefetchsize 512;
alter tablespace ts_stock_01 prefetchsize 512;
alter tablespace ts_stock_02 prefetchsize 512;
alter tablespace ts_stock_03 prefetchsize 512;
alter tablespace ts_stock_04 prefetchsize 512;
alter tablespace ts_stock_05 prefetchsize 512;
alter tablespace ts_stock_06 prefetchsize 512;
alter tablespace ts_stock_07 prefetchsize 512;
alter tablespace ts_stock_08 prefetchsize 512;
alter tablespace ts_stock_09 prefetchsize 512;
alter tablespace ts_stock_10 prefetchsize 512;
alter tablespace ts_stock_11 prefetchsize 512;
alter tablespace ts_stock_12 prefetchsize 512;
alter tablespace ts_stock_13 prefetchsize 512;
alter tablespace ts_stock_14 prefetchsize 512;
alter tablespace ts_stock_15 prefetchsize 512;
alter tablespace ts_stock_16 prefetchsize 512;
alter tablespace ts_stock_17 prefetchsize 512;
alter tablespace ts_stock_18 prefetchsize 512;
alter tablespace ts_stock_19 prefetchsize 512;
alter tablespace ts_stock_20 prefetchsize 512;
alter tablespace ts_stock_21 prefetchsize 512;
alter tablespace ts_stock_22 prefetchsize 512;
alter tablespace ts_stock_23 prefetchsize 512;
alter tablespace ts_stock_24 prefetchsize 512;
alter tablespace ts_stock_25 prefetchsize 512;
alter tablespace ts_stock_26 prefetchsize 512;
alter tablespace ts_stock_27 prefetchsize 512;
alter tablespace ts_stock_28 prefetchsize 512;
alter tablespace ts_ware_01 prefetchsize 512;
connect reset;
```

CRCONST_CUSTOMER.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER1 OFF;
ALTER TABLE CUSTOMER1 DROP CONSTRAINT CUSTOMER1CKC;
ALTER TABLE CUSTOMER1 ADD CONSTRAINT CUSTOMER1CKC CHECK (C_W_ID BETWEEN
1 AND 1900);
SET INTEGRITY FOR CUSTOMER1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER2 OFF;
ALTER TABLE CUSTOMER2 DROP CONSTRAINT CUSTOMER2CKC;
ALTER TABLE CUSTOMER2 ADD CONSTRAINT CUSTOMER2CKC CHECK (C_W_ID BETWEEN
1901 AND 3800);
SET INTEGRITY FOR CUSTOMER2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER3 OFF;
ALTER TABLE CUSTOMER3 DROP CONSTRAINT CUSTOMER3CKC;
ALTER TABLE CUSTOMER3 ADD CONSTRAINT CUSTOMER3CKC CHECK (C_W_ID BETWEEN
3801 AND 5700);
SET INTEGRITY FOR CUSTOMER3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER4 OFF;
ALTER TABLE CUSTOMER4 DROP CONSTRAINT CUSTOMER4CKC;
ALTER TABLE CUSTOMER4 ADD CONSTRAINT CUSTOMER4CKC CHECK (C_W_ID BETWEEN
5701 AND 7600);
SET INTEGRITY FOR CUSTOMER4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER5 OFF;
ALTER TABLE CUSTOMER5 DROP CONSTRAINT CUSTOMER5CKC;
ALTER TABLE CUSTOMER5 ADD CONSTRAINT CUSTOMER5CKC CHECK (C_W_ID BETWEEN
7601 AND 9500);
SET INTEGRITY FOR CUSTOMER5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER6 OFF;
ALTER TABLE CUSTOMER6 DROP CONSTRAINT CUSTOMER6CKC;
ALTER TABLE CUSTOMER6 ADD CONSTRAINT CUSTOMER6CKC CHECK (C_W_ID BETWEEN
9501 AND 11400);
SET INTEGRITY FOR CUSTOMER6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER7 OFF;
ALTER TABLE CUSTOMER7 DROP CONSTRAINT CUSTOMER7CKC;
ALTER TABLE CUSTOMER7 ADD CONSTRAINT CUSTOMER7CKC CHECK (C_W_ID BETWEEN
11401 AND 13300);
SET INTEGRITY FOR CUSTOMER7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER8 OFF;
ALTER TABLE CUSTOMER8 DROP CONSTRAINT CUSTOMER8CKC;
ALTER TABLE CUSTOMER8 ADD CONSTRAINT CUSTOMER8CKC CHECK (C_W_ID BETWEEN
13301 AND 15200);
SET INTEGRITY FOR CUSTOMER8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER9 OFF;
ALTER TABLE CUSTOMER9 DROP CONSTRAINT CUSTOMER9CKC;
ALTER TABLE CUSTOMER9 ADD CONSTRAINT CUSTOMER9CKC CHECK (C_W_ID BETWEEN
15201 AND 17100);
SET INTEGRITY FOR CUSTOMER9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER10 OFF;
ALTER TABLE CUSTOMER10 DROP CONSTRAINT CUSTOMER10CKC;
ALTER TABLE CUSTOMER10 ADD CONSTRAINT CUSTOMER10CKC CHECK (C_W_ID
BETWEEN 17101 AND 19000);
SET INTEGRITY FOR CUSTOMER10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER11 OFF;
ALTER TABLE CUSTOMER11 DROP CONSTRAINT CUSTOMER11CKC;
ALTER TABLE CUSTOMER11 ADD CONSTRAINT CUSTOMER11CKC CHECK (C_W_ID
BETWEEN 19001 AND 20900);
SET INTEGRITY FOR CUSTOMER11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER12 OFF;
ALTER TABLE CUSTOMER12 DROP CONSTRAINT CUSTOMER12CKC;
ALTER TABLE CUSTOMER12 ADD CONSTRAINT CUSTOMER12CKC CHECK (C_W_ID
BETWEEN 20901 AND 22800);
SET INTEGRITY FOR CUSTOMER12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER13 OFF;
ALTER TABLE CUSTOMER13 DROP CONSTRAINT CUSTOMER13CKC;
ALTER TABLE CUSTOMER13 ADD CONSTRAINT CUSTOMER13CKC CHECK (C_W_ID
BETWEEN 22801 AND 24700);
SET INTEGRITY FOR CUSTOMER13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER14 OFF;
ALTER TABLE CUSTOMER14 DROP CONSTRAINT CUSTOMER14CKC;
ALTER TABLE CUSTOMER14 ADD CONSTRAINT CUSTOMER14CKC CHECK (C_W_ID
BETWEEN 24701 AND 26600);
SET INTEGRITY FOR CUSTOMER14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER15 OFF;
ALTER TABLE CUSTOMER15 DROP CONSTRAINT CUSTOMER15CKC;
ALTER TABLE CUSTOMER15 ADD CONSTRAINT CUSTOMER15CKC CHECK (C_W_ID
BETWEEN 26601 AND 28500);
SET INTEGRITY FOR CUSTOMER15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER16 OFF;
ALTER TABLE CUSTOMER16 DROP CONSTRAINT CUSTOMER16CKC;
ALTER TABLE CUSTOMER16 ADD CONSTRAINT CUSTOMER16CKC CHECK (C_W_ID
BETWEEN 28501 AND 30400);
SET INTEGRITY FOR CUSTOMER16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER17 OFF;
ALTER TABLE CUSTOMER17 DROP CONSTRAINT CUSTOMER17CKC;
```

```
ALTER TABLE CUSTOMER17 ADD CONSTRAINT CUSTOMER17CKC CHECK (C_W_ID
BETWEEN 30401 AND 32300);
SET INTEGRITY FOR CUSTOMER17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER18 OFF;
ALTER TABLE CUSTOMER18 DROP CONSTRAINT CUSTOMER18CKC;
ALTER TABLE CUSTOMER18 ADD CONSTRAINT CUSTOMER18CKC CHECK (C_W_ID
BETWEEN 32301 AND 34200);
SET INTEGRITY FOR CUSTOMER18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER19 OFF;
ALTER TABLE CUSTOMER19 DROP CONSTRAINT CUSTOMER19CKC;
ALTER TABLE CUSTOMER19 ADD CONSTRAINT CUSTOMER19CKC CHECK (C_W_ID
BETWEEN 34201 AND 36100);
SET INTEGRITY FOR CUSTOMER19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER20 OFF;
ALTER TABLE CUSTOMER20 DROP CONSTRAINT CUSTOMER20CKC;
ALTER TABLE CUSTOMER20 ADD CONSTRAINT CUSTOMER20CKC CHECK (C_W_ID
BETWEEN 36101 AND 38000);
SET INTEGRITY FOR CUSTOMER20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER21 OFF;
ALTER TABLE CUSTOMER21 DROP CONSTRAINT CUSTOMER21CKC;
ALTER TABLE CUSTOMER21 ADD CONSTRAINT CUSTOMER21CKC CHECK (C_W_ID
BETWEEN 38001 AND 39900);
SET INTEGRITY FOR CUSTOMER21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER22 OFF;
ALTER TABLE CUSTOMER22 DROP CONSTRAINT CUSTOMER22CKC;
ALTER TABLE CUSTOMER22 ADD CONSTRAINT CUSTOMER22CKC CHECK (C_W_ID
BETWEEN 39901 AND 41800);
SET INTEGRITY FOR CUSTOMER22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER23 OFF;
ALTER TABLE CUSTOMER23 DROP CONSTRAINT CUSTOMER23CKC;
ALTER TABLE CUSTOMER23 ADD CONSTRAINT CUSTOMER23CKC CHECK (C_W_ID
BETWEEN 41801 AND 43700);
SET INTEGRITY FOR CUSTOMER23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER24 OFF;
ALTER TABLE CUSTOMER24 DROP CONSTRAINT CUSTOMER24CKC;
ALTER TABLE CUSTOMER24 ADD CONSTRAINT CUSTOMER24CKC CHECK (C_W_ID
BETWEEN 43701 AND 45600);
SET INTEGRITY FOR CUSTOMER24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER25 OFF;
ALTER TABLE CUSTOMER25 DROP CONSTRAINT CUSTOMER25CKC;
ALTER TABLE CUSTOMER25 ADD CONSTRAINT CUSTOMER25CKC CHECK (C_W_ID
BETWEEN 45601 AND 47500);
SET INTEGRITY FOR CUSTOMER25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER26 OFF;
ALTER TABLE CUSTOMER26 DROP CONSTRAINT CUSTOMER26CKC;
ALTER TABLE CUSTOMER26 ADD CONSTRAINT CUSTOMER26CKC CHECK (C_W_ID
BETWEEN 47501 AND 49400);
SET INTEGRITY FOR CUSTOMER26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER27 OFF;
ALTER TABLE CUSTOMER27 DROP CONSTRAINT CUSTOMER27CKC;
ALTER TABLE CUSTOMER27 ADD CONSTRAINT CUSTOMER27CKC CHECK (C_W_ID
BETWEEN 49401 AND 51300);
SET INTEGRITY FOR CUSTOMER27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR CUSTOMER28 OFF;
ALTER TABLE CUSTOMER28 DROP CONSTRAINT CUSTOMER28CKC;
ALTER TABLE CUSTOMER28 ADD CONSTRAINT CUSTOMER28CKC CHECK (C_W_ID >=
51301);
SET INTEGRITY FOR CUSTOMER28 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRONST_ORDER_LINE.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE1 OFF;
ALTER TABLE ORDER_LINE1 DROP CONSTRAINT ORDER_LINE1CKC;
ALTER TABLE ORDER_LINE1 ADD CONSTRAINT ORDER_LINE1CKC CHECK (OL_W_ID
BETWEEN 1 AND 1900);
SET INTEGRITY FOR ORDER_LINE1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE2 OFF;
ALTER TABLE ORDER_LINE2 DROP CONSTRAINT ORDER_LINE2CKC;
ALTER TABLE ORDER_LINE2 ADD CONSTRAINT ORDER_LINE2CKC CHECK (OL_W_ID
BETWEEN 1901 AND 3800);
SET INTEGRITY FOR ORDER_LINE2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE3 OFF;
ALTER TABLE ORDER_LINE3 DROP CONSTRAINT ORDER_LINE3CKC;
ALTER TABLE ORDER_LINE3 ADD CONSTRAINT ORDER_LINE3CKC CHECK (OL_W_ID
BETWEEN 3801 AND 5700);
SET INTEGRITY FOR ORDER_LINE3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE4 OFF;
ALTER TABLE ORDER_LINE4 DROP CONSTRAINT ORDER_LINE4CKC;
ALTER TABLE ORDER_LINE4 ADD CONSTRAINT ORDER_LINE4CKC CHECK (OL_W_ID
BETWEEN 5701 AND 7600);
SET INTEGRITY FOR ORDER_LINE4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE5 OFF;
ALTER TABLE ORDER_LINE5 DROP CONSTRAINT ORDER_LINE5CKC;
ALTER TABLE ORDER_LINE5 ADD CONSTRAINT ORDER_LINE5CKC CHECK (OL_W_ID
BETWEEN 7601 AND 9500);
SET INTEGRITY FOR ORDER_LINE5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE6 OFF;
ALTER TABLE ORDER_LINE6 DROP CONSTRAINT ORDER_LINE6CKC;
ALTER TABLE ORDER_LINE6 ADD CONSTRAINT ORDER_LINE6CKC CHECK (OL_W_ID
BETWEEN 9501 AND 11400);
SET INTEGRITY FOR ORDER_LINE6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE7 OFF;
ALTER TABLE ORDER_LINE7 DROP CONSTRAINT ORDER_LINE7CKC;
ALTER TABLE ORDER_LINE7 ADD CONSTRAINT ORDER_LINE7CKC CHECK (OL_W_ID
BETWEEN 11401 AND 13300);
SET INTEGRITY FOR ORDER_LINE7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE8 OFF;
ALTER TABLE ORDER_LINE8 DROP CONSTRAINT ORDER_LINE8CKC;
ALTER TABLE ORDER_LINE8 ADD CONSTRAINT ORDER_LINE8CKC CHECK (OL_W_ID
BETWEEN 13301 AND 15200);
SET INTEGRITY FOR ORDER_LINE8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE9 OFF;
ALTER TABLE ORDER_LINE9 DROP CONSTRAINT ORDER_LINE9CKC;
ALTER TABLE ORDER_LINE9 ADD CONSTRAINT ORDER_LINE9CKC CHECK (OL_W_ID
BETWEEN 15201 AND 17100);
SET INTEGRITY FOR ORDER_LINE9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE10 OFF;
ALTER TABLE ORDER_LINE10 DROP CONSTRAINT ORDER_LINE10CKC;
ALTER TABLE ORDER_LINE10 ADD CONSTRAINT ORDER_LINE10CKC CHECK (OL_W_ID
BETWEEN 17101 AND 19000);
SET INTEGRITY FOR ORDER_LINE10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE11 OFF;
ALTER TABLE ORDER_LINE11 DROP CONSTRAINT ORDER_LINE11CKC;
ALTER TABLE ORDER_LINE11 ADD CONSTRAINT ORDER_LINE11CKC CHECK (OL_W_ID
BETWEEN 19001 AND 20900);
SET INTEGRITY FOR ORDER_LINE11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE12 OFF;
ALTER TABLE ORDER_LINE12 DROP CONSTRAINT ORDER_LINE12CKC;
ALTER TABLE ORDER_LINE12 ADD CONSTRAINT ORDER_LINE12CKC CHECK (OL_W_ID
BETWEEN 20901 AND 22800);
SET INTEGRITY FOR ORDER_LINE12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE13 OFF;
ALTER TABLE ORDER_LINE13 DROP CONSTRAINT ORDER_LINE13CKC;
ALTER TABLE ORDER_LINE13 ADD CONSTRAINT ORDER_LINE13CKC CHECK (OL_W_ID
BETWEEN 22801 AND 24700);
SET INTEGRITY FOR ORDER_LINE13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE14 OFF;
ALTER TABLE ORDER_LINE14 DROP CONSTRAINT ORDER_LINE14CKC;
ALTER TABLE ORDER_LINE14 ADD CONSTRAINT ORDER_LINE14CKC CHECK (OL_W_ID
BETWEEN 24701 AND 26600);
SET INTEGRITY FOR ORDER_LINE14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE15 OFF;
ALTER TABLE ORDER_LINE15 DROP CONSTRAINT ORDER_LINE15CKC;
ALTER TABLE ORDER_LINE15 ADD CONSTRAINT ORDER_LINE15CKC CHECK (OL_W_ID
BETWEEN 26601 AND 28500);
SET INTEGRITY FOR ORDER_LINE15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE16 OFF;
ALTER TABLE ORDER_LINE16 DROP CONSTRAINT ORDER_LINE16CKC;
```

```
ALTER TABLE ORDER_LINE16 ADD CONSTRAINT ORDER_LINE16CKC CHECK (OL_W_ID
BETWEEN 28501 AND 30400);
SET INTEGRITY FOR ORDER_LINE16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE17 OFF;
ALTER TABLE ORDER_LINE17 DROP CONSTRAINT ORDER_LINE17CKC;
ALTER TABLE ORDER_LINE17 ADD CONSTRAINT ORDER_LINE17CKC CHECK (OL_W_ID
BETWEEN 30401 AND 32300);
SET INTEGRITY FOR ORDER_LINE17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE18 OFF;
ALTER TABLE ORDER_LINE18 DROP CONSTRAINT ORDER_LINE18CKC;
ALTER TABLE ORDER_LINE18 ADD CONSTRAINT ORDER_LINE18CKC CHECK (OL_W_ID
BETWEEN 32301 AND 34200);
SET INTEGRITY FOR ORDER_LINE18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE19 OFF;
ALTER TABLE ORDER_LINE19 DROP CONSTRAINT ORDER_LINE19CKC;
ALTER TABLE ORDER_LINE19 ADD CONSTRAINT ORDER_LINE19CKC CHECK (OL_W_ID
BETWEEN 34201 AND 36100);
SET INTEGRITY FOR ORDER_LINE19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE20 OFF;
ALTER TABLE ORDER_LINE20 DROP CONSTRAINT ORDER_LINE20CKC;
ALTER TABLE ORDER_LINE20 ADD CONSTRAINT ORDER_LINE20CKC CHECK (OL_W_ID
BETWEEN 36101 AND 38000);
SET INTEGRITY FOR ORDER_LINE20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE21 OFF;
ALTER TABLE ORDER_LINE21 DROP CONSTRAINT ORDER_LINE21CKC;
ALTER TABLE ORDER_LINE21 ADD CONSTRAINT ORDER_LINE21CKC CHECK (OL_W_ID
BETWEEN 38001 AND 39900);
SET INTEGRITY FOR ORDER_LINE21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE22 OFF;
ALTER TABLE ORDER_LINE22 DROP CONSTRAINT ORDER_LINE22CKC;
ALTER TABLE ORDER_LINE22 ADD CONSTRAINT ORDER_LINE22CKC CHECK (OL_W_ID
BETWEEN 39901 AND 41800);
SET INTEGRITY FOR ORDER_LINE22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE23 OFF;
ALTER TABLE ORDER_LINE23 DROP CONSTRAINT ORDER_LINE23CKC;
ALTER TABLE ORDER_LINE23 ADD CONSTRAINT ORDER_LINE23CKC CHECK (OL_W_ID
BETWEEN 41801 AND 43700);
SET INTEGRITY FOR ORDER_LINE23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE24 OFF;
ALTER TABLE ORDER_LINE24 DROP CONSTRAINT ORDER_LINE24CKC;
ALTER TABLE ORDER_LINE24 ADD CONSTRAINT ORDER_LINE24CKC CHECK (OL_W_ID
BETWEEN 43701 AND 45600);
SET INTEGRITY FOR ORDER_LINE24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE25 OFF;
ALTER TABLE ORDER_LINE25 DROP CONSTRAINT ORDER_LINE25CKC;
ALTER TABLE ORDER_LINE25 ADD CONSTRAINT ORDER_LINE25CKC CHECK (OL_W_ID
BETWEEN 45601 AND 47500);
SET INTEGRITY FOR ORDER_LINE25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE26 OFF;
ALTER TABLE ORDER_LINE26 DROP CONSTRAINT ORDER_LINE26CKC;
ALTER TABLE ORDER_LINE26 ADD CONSTRAINT ORDER_LINE26CKC CHECK (OL_W_ID
BETWEEN 47501 AND 49400);
SET INTEGRITY FOR ORDER_LINE26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE27 OFF;
ALTER TABLE ORDER_LINE27 DROP CONSTRAINT ORDER_LINE27CKC;
ALTER TABLE ORDER_LINE27 ADD CONSTRAINT ORDER_LINE27CKC CHECK (OL_W_ID
BETWEEN 49401 AND 51300);
SET INTEGRITY FOR ORDER_LINE27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDER_LINE28 OFF;
ALTER TABLE ORDER_LINE28 DROP CONSTRAINT ORDER_LINE28CKC;
ALTER TABLE ORDER_LINE28 ADD CONSTRAINT ORDER_LINE28CKC CHECK (OL_W_ID >=
51301);
SET INTEGRITY FOR ORDER_LINE28 ALL IMMEDIATE UNCHECKED;
connect reset;
```

CRONST_STOCK.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK1 OFF;
ALTER TABLE STOCK1 DROP CONSTRAINT STOCK1CKC;
ALTER TABLE STOCK1 ADD CONSTRAINT STOCK1CKC CHECK (S_W_ID BETWEEN 1 AND
1900);
SET INTEGRITY FOR STOCK1 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK2 OFF;
ALTER TABLE STOCK2 DROP CONSTRAINT STOCK2CKC;
ALTER TABLE STOCK2 ADD CONSTRAINT STOCK2CKC CHECK (S_W_ID BETWEEN 1901 AND
3800);
SET INTEGRITY FOR STOCK2 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK3 OFF;
ALTER TABLE STOCK3 DROP CONSTRAINT STOCK3CKC;
ALTER TABLE STOCK3 ADD CONSTRAINT STOCK3CKC CHECK (S_W_ID BETWEEN 3801 AND
5700);
SET INTEGRITY FOR STOCK3 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK4 OFF;
ALTER TABLE STOCK4 DROP CONSTRAINT STOCK4CKC;
ALTER TABLE STOCK4 ADD CONSTRAINT STOCK4CKC CHECK (S_W_ID BETWEEN 5701 AND
7600);
SET INTEGRITY FOR STOCK4 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK5 OFF;
ALTER TABLE STOCK5 DROP CONSTRAINT STOCK5CKC;
ALTER TABLE STOCK5 ADD CONSTRAINT STOCK5CKC CHECK (S_W_ID BETWEEN 7601 AND
9500);
SET INTEGRITY FOR STOCK5 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK6 OFF;
ALTER TABLE STOCK6 DROP CONSTRAINT STOCK6CKC;
ALTER TABLE STOCK6 ADD CONSTRAINT STOCK6CKC CHECK (S_W_ID BETWEEN 9501 AND
11400);
SET INTEGRITY FOR STOCK6 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK7 OFF;
ALTER TABLE STOCK7 DROP CONSTRAINT STOCK7CKC;
ALTER TABLE STOCK7 ADD CONSTRAINT STOCK7CKC CHECK (S_W_ID BETWEEN 11401
AND 13300);
SET INTEGRITY FOR STOCK7 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK8 OFF;
ALTER TABLE STOCK8 DROP CONSTRAINT STOCK8CKC;
ALTER TABLE STOCK8 ADD CONSTRAINT STOCK8CKC CHECK (S_W_ID BETWEEN 13301
AND 15200);
SET INTEGRITY FOR STOCK8 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK9 OFF;
ALTER TABLE STOCK9 DROP CONSTRAINT STOCK9CKC;
ALTER TABLE STOCK9 ADD CONSTRAINT STOCK9CKC CHECK (S_W_ID BETWEEN 15201
AND 17100);
SET INTEGRITY FOR STOCK9 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK10 OFF;
ALTER TABLE STOCK10 DROP CONSTRAINT STOCK10CKC;
ALTER TABLE STOCK10 ADD CONSTRAINT STOCK10CKC CHECK (S_W_ID BETWEEN 17101
AND 19000);
SET INTEGRITY FOR STOCK10 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK11 OFF;
ALTER TABLE STOCK11 DROP CONSTRAINT STOCK11CKC;
ALTER TABLE STOCK11 ADD CONSTRAINT STOCK11CKC CHECK (S_W_ID BETWEEN 19001
AND 20900);
SET INTEGRITY FOR STOCK11 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK12 OFF;
ALTER TABLE STOCK12 DROP CONSTRAINT STOCK12CKC;
ALTER TABLE STOCK12 ADD CONSTRAINT STOCK12CKC CHECK (S_W_ID BETWEEN 20901
AND 22800);
SET INTEGRITY FOR STOCK12 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK13 OFF;
ALTER TABLE STOCK13 DROP CONSTRAINT STOCK13CKC;
ALTER TABLE STOCK13 ADD CONSTRAINT STOCK13CKC CHECK (S_W_ID BETWEEN 22801
AND 24700);
SET INTEGRITY FOR STOCK13 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK14 OFF;
ALTER TABLE STOCK14 DROP CONSTRAINT STOCK14CKC;
ALTER TABLE STOCK14 ADD CONSTRAINT STOCK14CKC CHECK (S_W_ID BETWEEN 24701
AND 26600);
SET INTEGRITY FOR STOCK14 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK15 OFF;
ALTER TABLE STOCK15 DROP CONSTRAINT STOCK15CKC;
```

```
ALTER TABLE STOCK15 ADD CONSTRAINT STOCK15CKC CHECK (S_W_ID BETWEEN 26601
AND 28500);
SET INTEGRITY FOR STOCK15 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK16 OFF;
ALTER TABLE STOCK16 DROP CONSTRAINT STOCK16CKC;
ALTER TABLE STOCK16 ADD CONSTRAINT STOCK16CKC CHECK (S_W_ID BETWEEN 28501
AND 30400);
SET INTEGRITY FOR STOCK16 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK17 OFF;
ALTER TABLE STOCK17 DROP CONSTRAINT STOCK17CKC;
ALTER TABLE STOCK17 ADD CONSTRAINT STOCK17CKC CHECK (S_W_ID BETWEEN 30401
AND 32300);
SET INTEGRITY FOR STOCK17 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK18 OFF;
ALTER TABLE STOCK18 DROP CONSTRAINT STOCK18CKC;
ALTER TABLE STOCK18 ADD CONSTRAINT STOCK18CKC CHECK (S_W_ID BETWEEN 32301
AND 34200);
SET INTEGRITY FOR STOCK18 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK19 OFF;
ALTER TABLE STOCK19 DROP CONSTRAINT STOCK19CKC;
ALTER TABLE STOCK19 ADD CONSTRAINT STOCK19CKC CHECK (S_W_ID BETWEEN 34201
AND 36100);
SET INTEGRITY FOR STOCK19 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK20 OFF;
ALTER TABLE STOCK20 DROP CONSTRAINT STOCK20CKC;
ALTER TABLE STOCK20 ADD CONSTRAINT STOCK20CKC CHECK (S_W_ID BETWEEN 36101
AND 38000);
SET INTEGRITY FOR STOCK20 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK21 OFF;
ALTER TABLE STOCK21 DROP CONSTRAINT STOCK21CKC;
ALTER TABLE STOCK21 ADD CONSTRAINT STOCK21CKC CHECK (S_W_ID BETWEEN 38001
AND 39900);
SET INTEGRITY FOR STOCK21 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK22 OFF;
ALTER TABLE STOCK22 DROP CONSTRAINT STOCK22CKC;
ALTER TABLE STOCK22 ADD CONSTRAINT STOCK22CKC CHECK (S_W_ID BETWEEN 39901
AND 41800);
SET INTEGRITY FOR STOCK22 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK23 OFF;
ALTER TABLE STOCK23 DROP CONSTRAINT STOCK23CKC;
ALTER TABLE STOCK23 ADD CONSTRAINT STOCK23CKC CHECK (S_W_ID BETWEEN 41801
AND 43700);
SET INTEGRITY FOR STOCK23 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK24 OFF;
ALTER TABLE STOCK24 DROP CONSTRAINT STOCK24CKC;
ALTER TABLE STOCK24 ADD CONSTRAINT STOCK24CKC CHECK (S_W_ID BETWEEN 43701
AND 45600);
SET INTEGRITY FOR STOCK24 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK25 OFF;
ALTER TABLE STOCK25 DROP CONSTRAINT STOCK25CKC;
ALTER TABLE STOCK25 ADD CONSTRAINT STOCK25CKC CHECK (S_W_ID BETWEEN 45601
AND 47500);
SET INTEGRITY FOR STOCK25 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK26 OFF;
ALTER TABLE STOCK26 DROP CONSTRAINT STOCK26CKC;
ALTER TABLE STOCK26 ADD CONSTRAINT STOCK26CKC CHECK (S_W_ID BETWEEN 47501
AND 49400);
SET INTEGRITY FOR STOCK26 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK27 OFF;
ALTER TABLE STOCK27 DROP CONSTRAINT STOCK27CKC;
ALTER TABLE STOCK27 ADD CONSTRAINT STOCK27CKC CHECK (S_W_ID BETWEEN 49401
AND 51300);
SET INTEGRITY FOR STOCK27 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR STOCK28 OFF;
ALTER TABLE STOCK28 DROP CONSTRAINT STOCK28CKC;
ALTER TABLE STOCK28 ADD CONSTRAINT STOCK28CKC CHECK (S_W_ID >= 51301);
SET INTEGRITY FOR STOCK28 ALL IMMEDIATE UNCHECKED;

connect reset;
```

CRIDX_CUST_IDXB.ddl

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB1;
CREATE INDEX CUST_IDXB1
        ON CUSTOMER1(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB2;
CREATE INDEX CUST_IDXB2
        ON CUSTOMER2(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB3;
CREATE INDEX CUST_IDXB3
        ON CUSTOMER3(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB4;
CREATE INDEX CUST_IDXB4
        ON CUSTOMER4(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB5;
CREATE INDEX CUST_IDXB5
        ON CUSTOMER5(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB6;
CREATE INDEX CUST_IDXB6
        ON CUSTOMER6(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB7;
CREATE INDEX CUST_IDXB7
        ON CUSTOMER7(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB8;
CREATE INDEX CUST_IDXB8
        ON CUSTOMER8(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB9;
CREATE INDEX CUST_IDXB9
        ON CUSTOMER9(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB10;
CREATE INDEX CUST_IDXB10
        ON CUSTOMER10(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB11;
CREATE INDEX CUST_IDXB11
        ON CUSTOMER11(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB12;
CREATE INDEX CUST_IDXB12
        ON CUSTOMER12(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB13;
CREATE INDEX CUST_IDXB13
        ON CUSTOMER13(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB14;
CREATE INDEX CUST_IDXB14
        ON CUSTOMER14(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB15;
CREATE INDEX CUST_IDXB15
        ON CUSTOMER15(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB16;
CREATE INDEX CUST_IDXB16
        ON CUSTOMER16(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB17;
CREATE INDEX CUST_IDXB17
        ON CUSTOMER17(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB18;
CREATE INDEX CUST_IDXB18
        ON CUSTOMER18(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB19;
CREATE INDEX CUST_IDXB19
        ON CUSTOMER19(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
```

```
connect to TPCC in share mode;
DROP INDEX CUST_IDXB20;
CREATE INDEX CUST_IDXB20
        ON CUSTOMER20(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB21;
CREATE INDEX CUST_IDXB21
        ON CUSTOMER21(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB22;
CREATE INDEX CUST_IDXB22
        ON CUSTOMER22(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB23;
CREATE INDEX CUST_IDXB23
        ON CUSTOMER23(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB24;
CREATE INDEX CUST_IDXB24
        ON CUSTOMER24(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB25;
CREATE INDEX CUST_IDXB25
        ON CUSTOMER25(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB26;
CREATE INDEX CUST_IDXB26
        ON CUSTOMER26(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB27;
CREATE INDEX CUST_IDXB27
        ON CUSTOMER27(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;
connect reset;
connect to TPCC in share mode;
DROP INDEX CUST_IDXB28;
CREATE INDEX CUST_IDXB28
        ON CUSTOMER28(C_LAST, C_W_ID, C_D_ID, C_FIRST, C_ID) PCTFREE 0;

connect reset;
```

CRIDX_NU_ORD_IDXA.ddl

```
connect to TPCC in share mode;
DROP INDEX NU_ORD_IDXA;
CREATE UNIQUE INDEX NU_ORD_IDXA
        ON NEW_ORDER(NO_W_ID, NO_D_ID, NO_O_ID) PCTFREE 20 LEVEL2 PCTFREE
20 PAGE SPLIT HIGH;

connect reset;
```

CRIDX_ORDR_IDXB.ddl

```
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB;
CREATE INDEX ORDR_IDXB
        ON ORDERS(O_C_ID, O_W_ID, O_D_ID, O_ID DESC) PCTFREE 20 LEVEL2
PCTFREE 20;

connect reset;
```

CRTB_CUSTOMER.ddl

```
connect to TPCC in share mode;
DROP TABLE CUSTOMER1;
CREATE TABLE CUSTOMER1
        (
         C_ID          INTEGER      NOT NULL,
         C_STATE       CHAR(2)      NOT NULL,
         C_ZIP         CHAR(9)      NOT NULL,
         C_PHONE       CHAR(16)     NOT NULL,
         C_SINCE       BIGINT       NOT NULL,
         C_CREDIT_LIM  BIGINT       NOT NULL,
         C_MIDDLE      CHAR(2)      NOT NULL,
         C_CREDIT      CHAR(2)      NOT NULL,
         C_DISCOUNT    INTEGER      NOT NULL,
         C_DATA        VARCHAR(500) NOT NULL,
         C_LAST        VARCHAR(16)  NOT NULL,
         C_FIRST       VARCHAR(16)  NOT NULL,
         C_STREET_1    VARCHAR(20)  NOT NULL,
         C_STREET_2    VARCHAR(20)  NOT NULL,
         C_CITY        VARCHAR(20)  NOT NULL,
         C_D_ID        SMALLINT     NOT NULL,
         C_W_ID        INTEGER      NOT NULL,
         C_DELIVERY_CNT INTEGER     NOT NULL,
         C_BALANCE     BIGINT       NOT NULL,
         C_YTD_PAYMENT BIGINT       NOT NULL,
         C_PAYMENT_CNT INTEGER      NOT NULL
        )
        IN ts_customer_01
        INDEX IN is_customer_01
        ORGANIZE BY KEY SEQUENCE (
         C_ID STARTING FROM 1 ENDING AT 3000,
         C_W_ID STARTING FROM 1 ENDING AT 1900,
         C_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER2;
CREATE TABLE CUSTOMER2
        (
         C_ID          INTEGER      NOT NULL,
         C_STATE       CHAR(2)      NOT NULL,
         C_ZIP         CHAR(9)      NOT NULL,
         C_PHONE       CHAR(16)     NOT NULL,
         C_SINCE       BIGINT       NOT NULL,
         C_CREDIT_LIM  BIGINT       NOT NULL,
         C_MIDDLE      CHAR(2)      NOT NULL,
         C_CREDIT      CHAR(2)      NOT NULL,
         C_DISCOUNT    INTEGER      NOT NULL,
         C_DATA        VARCHAR(500) NOT NULL,
         C_LAST        VARCHAR(16)  NOT NULL,
         C_FIRST       VARCHAR(16)  NOT NULL,
         C_STREET_1    VARCHAR(20)  NOT NULL,
         C_STREET_2    VARCHAR(20)  NOT NULL,
         C_CITY        VARCHAR(20)  NOT NULL,
         C_D_ID        SMALLINT     NOT NULL,
         C_W_ID        INTEGER      NOT NULL,
         C_DELIVERY_CNT INTEGER     NOT NULL,
         C_BALANCE     BIGINT       NOT NULL,
         C_YTD_PAYMENT BIGINT       NOT NULL,
         C_PAYMENT_CNT INTEGER      NOT NULL
        )
        IN ts_customer_02
        INDEX IN is_customer_02
        ORGANIZE BY KEY SEQUENCE (
         C_ID STARTING FROM 1 ENDING AT 3000,
         C_W_ID STARTING FROM 1901 ENDING AT 3800,
         C_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER3;
CREATE TABLE CUSTOMER3
        (
         C_ID          INTEGER      NOT NULL,
         C_STATE       CHAR(2)      NOT NULL,
         C_ZIP         CHAR(9)      NOT NULL,
         C_PHONE       CHAR(16)     NOT NULL,
         C_SINCE       BIGINT       NOT NULL,
         C_CREDIT_LIM  BIGINT       NOT NULL,
         C_MIDDLE      CHAR(2)      NOT NULL,
         C_CREDIT      CHAR(2)      NOT NULL,
         C_DISCOUNT    INTEGER      NOT NULL,
         C_DATA        VARCHAR(500) NOT NULL,
         C_LAST        VARCHAR(16)  NOT NULL,
         C_FIRST       VARCHAR(16)  NOT NULL,
         C_STREET_1    VARCHAR(20)  NOT NULL,
         C_STREET_2    VARCHAR(20)  NOT NULL,
         C_CITY        VARCHAR(20)  NOT NULL,
         C_D_ID        SMALLINT     NOT NULL,
         C_W_ID        INTEGER      NOT NULL,
         C_DELIVERY_CNT INTEGER     NOT NULL,
         C_BALANCE     BIGINT       NOT NULL,
         C_YTD_PAYMENT BIGINT       NOT NULL,
         C_PAYMENT_CNT INTEGER      NOT NULL
        )
        IN ts_customer_03
        INDEX IN is_customer_03
        ORGANIZE BY KEY SEQUENCE (
         C_ID STARTING FROM 1 ENDING AT 3000,
         C_W_ID STARTING FROM 3801 ENDING AT 5700,
         C_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER4;
CREATE TABLE CUSTOMER4
        (
         C_ID          INTEGER      NOT NULL,
         C_STATE       CHAR(2)      NOT NULL,
         C_ZIP         CHAR(9)      NOT NULL,
         C_PHONE       CHAR(16)     NOT NULL,
         C_SINCE       BIGINT       NOT NULL,
         C_CREDIT_LIM  BIGINT       NOT NULL,
         C_MIDDLE      CHAR(2)      NOT NULL,
         C_CREDIT      CHAR(2)      NOT NULL,
         C_DISCOUNT    INTEGER      NOT NULL,
```

```
            C_DATA        VARCHAR(500)  NOT NULL,
            C_LAST        VARCHAR(16)   NOT NULL,
            C_FIRST       VARCHAR(16)   NOT NULL,
            C_STREET_1    VARCHAR(20)   NOT NULL,
            C_STREET_2    VARCHAR(20)   NOT NULL,
            C_CITY        VARCHAR(20)   NOT NULL,
            C_D_ID        SMALLINT      NOT NULL,
            C_W_ID        INTEGER       NOT NULL,
            C_DELIVERY_CNT INTEGER      NOT NULL,
            C_BALANCE     BIGINT        NOT NULL,
            C_YTD_PAYMENT BIGINT        NOT NULL,
            C_PAYMENT_CNT INTEGER       NOT NULL
            )
            IN ts_customer_04
            INDEX IN is_customer_04
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 5701 ENDING AT 7600,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER5;
CREATE TABLE CUSTOMER5
            (
            C_ID          INTEGER       NOT NULL,
            C_STATE       CHAR(2)       NOT NULL,
            C_ZIP         CHAR(9)       NOT NULL,
            C_PHONE       CHAR(16)      NOT NULL,
            C_SINCE       BIGINT        NOT NULL,
            C_CREDIT_LIM  BIGINT        NOT NULL,
            C_MIDDLE      CHAR(2)       NOT NULL,
            C_CREDIT      CHAR(2)       NOT NULL,
            C_DISCOUNT    INTEGER       NOT NULL,
            C_DATA        VARCHAR(500)  NOT NULL,
            C_LAST        VARCHAR(16)   NOT NULL,
            C_FIRST       VARCHAR(16)   NOT NULL,
            C_STREET_1    VARCHAR(20)   NOT NULL,
            C_STREET_2    VARCHAR(20)   NOT NULL,
            C_CITY        VARCHAR(20)   NOT NULL,
            C_D_ID        SMALLINT      NOT NULL,
            C_W_ID        INTEGER       NOT NULL,
            C_DELIVERY_CNT INTEGER      NOT NULL,
            C_BALANCE     BIGINT        NOT NULL,
            C_YTD_PAYMENT BIGINT        NOT NULL,
            C_PAYMENT_CNT INTEGER       NOT NULL
            )
            IN ts_customer_05
            INDEX IN is_customer_05
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 7601 ENDING AT 9500,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER6;
CREATE TABLE CUSTOMER6
            (
            C_ID          INTEGER       NOT NULL,
            C_STATE       CHAR(2)       NOT NULL,
            C_ZIP         CHAR(9)       NOT NULL,
            C_PHONE       CHAR(16)      NOT NULL,
            C_SINCE       BIGINT        NOT NULL,
            C_CREDIT_LIM  BIGINT        NOT NULL,
            C_MIDDLE      CHAR(2)       NOT NULL,
            C_CREDIT      CHAR(2)       NOT NULL,
            C_DISCOUNT    INTEGER       NOT NULL,
            C_DATA        VARCHAR(500)  NOT NULL,
            C_LAST        VARCHAR(16)   NOT NULL,
            C_FIRST       VARCHAR(16)   NOT NULL,
            C_STREET_1    VARCHAR(20)   NOT NULL,
            C_STREET_2    VARCHAR(20)   NOT NULL,
            C_CITY        VARCHAR(20)   NOT NULL,
            C_D_ID        SMALLINT      NOT NULL,
            C_W_ID        INTEGER       NOT NULL,
            C_DELIVERY_CNT INTEGER      NOT NULL,
            C_BALANCE     BIGINT        NOT NULL,
            C_YTD_PAYMENT BIGINT        NOT NULL,
            C_PAYMENT_CNT INTEGER       NOT NULL
            )
            IN ts_customer_06
            INDEX IN is_customer_06
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 9501 ENDING AT 11400,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER7;
CREATE TABLE CUSTOMER7
            (
            C_ID          INTEGER       NOT NULL,
            C_STATE       CHAR(2)       NOT NULL,
            C_ZIP         CHAR(9)       NOT NULL,
            C_PHONE       CHAR(16)      NOT NULL,
            C_SINCE       BIGINT        NOT NULL,
            C_CREDIT_LIM  BIGINT        NOT NULL,
            C_MIDDLE      CHAR(2)       NOT NULL,
            C_CREDIT      CHAR(2)       NOT NULL,
            C_DISCOUNT    INTEGER       NOT NULL,
            C_DATA        VARCHAR(500)  NOT NULL,
            C_LAST        VARCHAR(16)   NOT NULL,
            C_FIRST       VARCHAR(16)   NOT NULL,
            C_STREET_1    VARCHAR(20)   NOT NULL,
            C_STREET_2    VARCHAR(20)   NOT NULL,
            C_CITY        VARCHAR(20)   NOT NULL,
            C_D_ID        SMALLINT      NOT NULL,
            C_W_ID        INTEGER       NOT NULL,
            C_DELIVERY_CNT INTEGER      NOT NULL,
            C_BALANCE     BIGINT        NOT NULL,
            C_YTD_PAYMENT BIGINT        NOT NULL,
            C_PAYMENT_CNT INTEGER       NOT NULL
            )
            IN ts_customer_07
            INDEX IN is_customer_07
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 11401 ENDING AT 13300,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER8;
CREATE TABLE CUSTOMER8
            (
            C_ID          INTEGER       NOT NULL,
            C_STATE       CHAR(2)       NOT NULL,
            C_ZIP         CHAR(9)       NOT NULL,
            C_PHONE       CHAR(16)      NOT NULL,
            C_SINCE       BIGINT        NOT NULL,
            C_CREDIT_LIM  BIGINT        NOT NULL,
            C_MIDDLE      CHAR(2)       NOT NULL,
            C_CREDIT      CHAR(2)       NOT NULL,
            C_DISCOUNT    INTEGER       NOT NULL,
            C_DATA        VARCHAR(500)  NOT NULL,
            C_LAST        VARCHAR(16)   NOT NULL,
            C_FIRST       VARCHAR(16)   NOT NULL,
            C_STREET_1    VARCHAR(20)   NOT NULL,
            C_STREET_2    VARCHAR(20)   NOT NULL,
            C_CITY        VARCHAR(20)   NOT NULL,
            C_D_ID        SMALLINT      NOT NULL,
            C_W_ID        INTEGER       NOT NULL,
            C_DELIVERY_CNT INTEGER      NOT NULL,
            C_BALANCE     BIGINT        NOT NULL,
            C_YTD_PAYMENT BIGINT        NOT NULL,
            C_PAYMENT_CNT INTEGER       NOT NULL
            )
            IN ts_customer_08
            INDEX IN is_customer_08
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 13301 ENDING AT 15200,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER9;
CREATE TABLE CUSTOMER9
            (
            C_ID          INTEGER       NOT NULL,
            C_STATE       CHAR(2)       NOT NULL,
            C_ZIP         CHAR(9)       NOT NULL,
            C_PHONE       CHAR(16)      NOT NULL,
            C_SINCE       BIGINT        NOT NULL,
            C_CREDIT_LIM  BIGINT        NOT NULL,
            C_MIDDLE      CHAR(2)       NOT NULL,
            C_CREDIT      CHAR(2)       NOT NULL,
            C_DISCOUNT    INTEGER       NOT NULL,
            C_DATA        VARCHAR(500)  NOT NULL,
            C_LAST        VARCHAR(16)   NOT NULL,
            C_FIRST       VARCHAR(16)   NOT NULL,
            C_STREET_1    VARCHAR(20)   NOT NULL,
            C_STREET_2    VARCHAR(20)   NOT NULL,
            C_CITY        VARCHAR(20)   NOT NULL,
            C_D_ID        SMALLINT      NOT NULL,
            C_W_ID        INTEGER       NOT NULL,
            C_DELIVERY_CNT INTEGER      NOT NULL,
            C_BALANCE     BIGINT        NOT NULL,
            C_YTD_PAYMENT BIGINT        NOT NULL,
            C_PAYMENT_CNT INTEGER       NOT NULL
            )
            IN ts_customer_09
            INDEX IN is_customer_09
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 15201 ENDING AT 17100,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
```

```
DROP TABLE CUSTOMER10;
CREATE TABLE CUSTOMER10
            (
             C_ID          INTEGER      NOT NULL,
             C_STATE       CHAR(2)      NOT NULL,
             C_ZIP         CHAR(9)      NOT NULL,
             C_PHONE       CHAR(16)     NOT NULL,
             C_SINCE       BIGINT       NOT NULL,
             C_CREDIT_LIM  BIGINT       NOT NULL,
             C_MIDDLE      CHAR(2)      NOT NULL,
             C_CREDIT      CHAR(2)      NOT NULL,
             C_DISCOUNT    INTEGER      NOT NULL,
             C_DATA        VARCHAR(500) NOT NULL,
             C_LAST        VARCHAR(16)  NOT NULL,
             C_FIRST       VARCHAR(16)  NOT NULL,
             C_STREET_1    VARCHAR(20)  NOT NULL,
             C_STREET_2    VARCHAR(20)  NOT NULL,
             C_CITY        VARCHAR(20)  NOT NULL,
             C_D_ID        SMALLINT     NOT NULL,
             C_W_ID        INTEGER      NOT NULL,
             C_DELIVERY_CNT INTEGER     NOT NULL,
             C_BALANCE     BIGINT       NOT NULL,
             C_YTD_PAYMENT BIGINT       NOT NULL,
             C_PAYMENT_CNT INTEGER      NOT NULL
            )
            IN ts_customer_10
            INDEX IN is_customer_10
            ORGANIZE BY KEY SEQUENCE (
             C_ID STARTING FROM 1 ENDING AT 3000,
             C_W_ID STARTING FROM 17101 ENDING AT 19000,
             C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER11;
CREATE TABLE CUSTOMER11
            (
             C_ID          INTEGER      NOT NULL,
             C_STATE       CHAR(2)      NOT NULL,
             C_ZIP         CHAR(9)      NOT NULL,
             C_PHONE       CHAR(16)     NOT NULL,
             C_SINCE       BIGINT       NOT NULL,
             C_CREDIT_LIM  BIGINT       NOT NULL,
             C_MIDDLE      CHAR(2)      NOT NULL,
             C_CREDIT      CHAR(2)      NOT NULL,
             C_DISCOUNT    INTEGER      NOT NULL,
             C_DATA        VARCHAR(500) NOT NULL,
             C_LAST        VARCHAR(16)  NOT NULL,
             C_FIRST       VARCHAR(16)  NOT NULL,
             C_STREET_1    VARCHAR(20)  NOT NULL,
             C_STREET_2    VARCHAR(20)  NOT NULL,
             C_CITY        VARCHAR(20)  NOT NULL,
             C_D_ID        SMALLINT     NOT NULL,
             C_W_ID        INTEGER      NOT NULL,
             C_DELIVERY_CNT INTEGER     NOT NULL,
             C_BALANCE     BIGINT       NOT NULL,
             C_YTD_PAYMENT BIGINT       NOT NULL,
             C_PAYMENT_CNT INTEGER      NOT NULL
            )
            IN ts_customer_11
            INDEX IN is_customer_11
            ORGANIZE BY KEY SEQUENCE (
             C_ID STARTING FROM 1 ENDING AT 3000,
             C_W_ID STARTING FROM 19001 ENDING AT 20900,
             C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER12;
CREATE TABLE CUSTOMER12
            (
             C_ID          INTEGER      NOT NULL,
             C_STATE       CHAR(2)      NOT NULL,
             C_ZIP         CHAR(9)      NOT NULL,
             C_PHONE       CHAR(16)     NOT NULL,
             C_SINCE       BIGINT       NOT NULL,
             C_CREDIT_LIM  BIGINT       NOT NULL,
             C_MIDDLE      CHAR(2)      NOT NULL,
             C_CREDIT      CHAR(2)      NOT NULL,
             C_DISCOUNT    INTEGER      NOT NULL,
             C_DATA        VARCHAR(500) NOT NULL,
             C_LAST        VARCHAR(16)  NOT NULL,
             C_FIRST       VARCHAR(16)  NOT NULL,
             C_STREET_1    VARCHAR(20)  NOT NULL,
             C_STREET_2    VARCHAR(20)  NOT NULL,
             C_CITY        VARCHAR(20)  NOT NULL,
             C_D_ID        SMALLINT     NOT NULL,
             C_W_ID        INTEGER      NOT NULL,
             C_DELIVERY_CNT INTEGER     NOT NULL,
             C_BALANCE     BIGINT       NOT NULL,
             C_YTD_PAYMENT BIGINT       NOT NULL,
             C_PAYMENT_CNT INTEGER      NOT NULL
            )
            IN ts_customer_12
            INDEX IN is_customer_12
            ORGANIZE BY KEY SEQUENCE (
             C_ID STARTING FROM 1 ENDING AT 3000,
             C_W_ID STARTING FROM 20901 ENDING AT 22800,
             C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER13;
CREATE TABLE CUSTOMER13
            (
             C_ID          INTEGER      NOT NULL,
             C_STATE       CHAR(2)      NOT NULL,
             C_ZIP         CHAR(9)      NOT NULL,
             C_PHONE       CHAR(16)     NOT NULL,
             C_SINCE       BIGINT       NOT NULL,
             C_CREDIT_LIM  BIGINT       NOT NULL,
             C_MIDDLE      CHAR(2)      NOT NULL,
             C_CREDIT      CHAR(2)      NOT NULL,
             C_DISCOUNT    INTEGER      NOT NULL,
             C_DATA        VARCHAR(500) NOT NULL,
             C_LAST        VARCHAR(16)  NOT NULL,
             C_FIRST       VARCHAR(16)  NOT NULL,
             C_STREET_1    VARCHAR(20)  NOT NULL,
             C_STREET_2    VARCHAR(20)  NOT NULL,
             C_CITY        VARCHAR(20)  NOT NULL,
             C_D_ID        SMALLINT     NOT NULL,
             C_W_ID        INTEGER      NOT NULL,
             C_DELIVERY_CNT INTEGER     NOT NULL,
             C_BALANCE     BIGINT       NOT NULL,
             C_YTD_PAYMENT BIGINT       NOT NULL,
             C_PAYMENT_CNT INTEGER      NOT NULL
            )
            IN ts_customer_13
            INDEX IN is_customer_13
            ORGANIZE BY KEY SEQUENCE (
             C_ID STARTING FROM 1 ENDING AT 3000,
             C_W_ID STARTING FROM 22801 ENDING AT 24700,
             C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER14;
CREATE TABLE CUSTOMER14
            (
             C_ID          INTEGER      NOT NULL,
             C_STATE       CHAR(2)      NOT NULL,
             C_ZIP         CHAR(9)      NOT NULL,
             C_PHONE       CHAR(16)     NOT NULL,
             C_SINCE       BIGINT       NOT NULL,
             C_CREDIT_LIM  BIGINT       NOT NULL,
             C_MIDDLE      CHAR(2)      NOT NULL,
             C_CREDIT      CHAR(2)      NOT NULL,
             C_DISCOUNT    INTEGER      NOT NULL,
             C_DATA        VARCHAR(500) NOT NULL,
             C_LAST        VARCHAR(16)  NOT NULL,
             C_FIRST       VARCHAR(16)  NOT NULL,
             C_STREET_1    VARCHAR(20)  NOT NULL,
             C_STREET_2    VARCHAR(20)  NOT NULL,
             C_CITY        VARCHAR(20)  NOT NULL,
             C_D_ID        SMALLINT     NOT NULL,
             C_W_ID        INTEGER      NOT NULL,
             C_DELIVERY_CNT INTEGER     NOT NULL,
             C_BALANCE     BIGINT       NOT NULL,
             C_YTD_PAYMENT BIGINT       NOT NULL,
             C_PAYMENT_CNT INTEGER      NOT NULL
            )
            IN ts_customer_14
            INDEX IN is_customer_14
            ORGANIZE BY KEY SEQUENCE (
             C_ID STARTING FROM 1 ENDING AT 3000,
             C_W_ID STARTING FROM 24701 ENDING AT 26600,
             C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER15;
CREATE TABLE CUSTOMER15
            (
             C_ID          INTEGER      NOT NULL,
             C_STATE       CHAR(2)      NOT NULL,
             C_ZIP         CHAR(9)      NOT NULL,
             C_PHONE       CHAR(16)     NOT NULL,
             C_SINCE       BIGINT       NOT NULL,
             C_CREDIT_LIM  BIGINT       NOT NULL,
             C_MIDDLE      CHAR(2)      NOT NULL,
             C_CREDIT      CHAR(2)      NOT NULL,
             C_DISCOUNT    INTEGER      NOT NULL,
             C_DATA        VARCHAR(500) NOT NULL,
             C_LAST        VARCHAR(16)  NOT NULL,
             C_FIRST       VARCHAR(16)  NOT NULL,
             C_STREET_1    VARCHAR(20)  NOT NULL,
             C_STREET_2    VARCHAR(20)  NOT NULL,
             C_CITY        VARCHAR(20)  NOT NULL,
             C_D_ID        SMALLINT     NOT NULL,
             C_W_ID        INTEGER      NOT NULL,
             C_DELIVERY_CNT INTEGER     NOT NULL,
             C_BALANCE     BIGINT       NOT NULL,
             C_YTD_PAYMENT BIGINT       NOT NULL,
```

```
            C_PAYMENT_CNT   INTEGER      NOT NULL
            )
            IN ts_customer_15
            INDEX IN is_customer_15
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 26601 ENDING AT 28500,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER16;
CREATE TABLE CUSTOMER16
            (
            C_ID          INTEGER      NOT NULL,
            C_STATE       CHAR(2)      NOT NULL,
            C_ZIP         CHAR(9)      NOT NULL,
            C_PHONE       CHAR(16)     NOT NULL,
            C_SINCE       BIGINT       NOT NULL,
            C_CREDIT_LIM  BIGINT       NOT NULL,
            C_MIDDLE      CHAR(2)      NOT NULL,
            C_CREDIT      CHAR(2)      NOT NULL,
            C_DISCOUNT    INTEGER      NOT NULL,
            C_DATA        VARCHAR(500) NOT NULL,
            C_LAST        VARCHAR(16)  NOT NULL,
            C_FIRST       VARCHAR(16)  NOT NULL,
            C_STREET_1    VARCHAR(20)  NOT NULL,
            C_STREET_2    VARCHAR(20)  NOT NULL,
            C_CITY        VARCHAR(20)  NOT NULL,
            C_D_ID        SMALLINT     NOT NULL,
            C_W_ID        INTEGER      NOT NULL,
            C_DELIVERY_CNT INTEGER     NOT NULL,
            C_BALANCE     BIGINT       NOT NULL,
            C_YTD_PAYMENT  BIGINT      NOT NULL,
            C_PAYMENT_CNT  INTEGER     NOT NULL
            )
            IN ts_customer_16
            INDEX IN is_customer_16
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 28501 ENDING AT 30400,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER17;
CREATE TABLE CUSTOMER17
            (
            C_ID          INTEGER      NOT NULL,
            C_STATE       CHAR(2)      NOT NULL,
            C_ZIP         CHAR(9)      NOT NULL,
            C_PHONE       CHAR(16)     NOT NULL,
            C_SINCE       BIGINT       NOT NULL,
            C_CREDIT_LIM  BIGINT       NOT NULL,
            C_MIDDLE      CHAR(2)      NOT NULL,
            C_CREDIT      CHAR(2)      NOT NULL,
            C_DISCOUNT    INTEGER      NOT NULL,
            C_DATA        VARCHAR(500) NOT NULL,
            C_LAST        VARCHAR(16)  NOT NULL,
            C_FIRST       VARCHAR(16)  NOT NULL,
            C_STREET_1    VARCHAR(20)  NOT NULL,
            C_STREET_2    VARCHAR(20)  NOT NULL,
            C_CITY        VARCHAR(20)  NOT NULL,
            C_D_ID        SMALLINT     NOT NULL,
            C_W_ID        INTEGER      NOT NULL,
            C_DELIVERY_CNT INTEGER     NOT NULL,
            C_BALANCE     BIGINT       NOT NULL,
            C_YTD_PAYMENT  BIGINT      NOT NULL,
            C_PAYMENT_CNT  INTEGER     NOT NULL
            )
            IN ts_customer_17
            INDEX IN is_customer_17
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 30401 ENDING AT 32300,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER18;
CREATE TABLE CUSTOMER18
            (
            C_ID          INTEGER      NOT NULL,
            C_STATE       CHAR(2)      NOT NULL,
            C_ZIP         CHAR(9)      NOT NULL,
            C_PHONE       CHAR(16)     NOT NULL,
            C_SINCE       BIGINT       NOT NULL,
            C_CREDIT_LIM  BIGINT       NOT NULL,
            C_MIDDLE      CHAR(2)      NOT NULL,
            C_CREDIT      CHAR(2)      NOT NULL,
            C_DISCOUNT    INTEGER      NOT NULL,
            C_DATA        VARCHAR(500) NOT NULL,
            C_LAST        VARCHAR(16)  NOT NULL,
            C_FIRST       VARCHAR(16)  NOT NULL,
            C_STREET_1    VARCHAR(20)  NOT NULL,
            C_STREET_2    VARCHAR(20)  NOT NULL,
            C_CITY        VARCHAR(20)  NOT NULL,
            C_D_ID        SMALLINT     NOT NULL,
            C_W_ID        INTEGER      NOT NULL,
            C_DELIVERY_CNT INTEGER     NOT NULL,
            C_BALANCE     BIGINT       NOT NULL,
            C_YTD_PAYMENT  BIGINT      NOT NULL,
            C_PAYMENT_CNT  INTEGER     NOT NULL
            )
            IN ts_customer_18
            INDEX IN is_customer_18
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 32301 ENDING AT 34200,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER19;
CREATE TABLE CUSTOMER19
            (
            C_ID          INTEGER      NOT NULL,
            C_STATE       CHAR(2)      NOT NULL,
            C_ZIP         CHAR(9)      NOT NULL,
            C_PHONE       CHAR(16)     NOT NULL,
            C_SINCE       BIGINT       NOT NULL,
            C_CREDIT_LIM  BIGINT       NOT NULL,
            C_MIDDLE      CHAR(2)      NOT NULL,
            C_CREDIT      CHAR(2)      NOT NULL,
            C_DISCOUNT    INTEGER      NOT NULL,
            C_DATA        VARCHAR(500) NOT NULL,
            C_LAST        VARCHAR(16)  NOT NULL,
            C_FIRST       VARCHAR(16)  NOT NULL,
            C_STREET_1    VARCHAR(20)  NOT NULL,
            C_STREET_2    VARCHAR(20)  NOT NULL,
            C_CITY        VARCHAR(20)  NOT NULL,
            C_D_ID        SMALLINT     NOT NULL,
            C_W_ID        INTEGER      NOT NULL,
            C_DELIVERY_CNT INTEGER     NOT NULL,
            C_BALANCE     BIGINT       NOT NULL,
            C_YTD_PAYMENT  BIGINT      NOT NULL,
            C_PAYMENT_CNT  INTEGER     NOT NULL
            )
            IN ts_customer_19
            INDEX IN is_customer_19
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 34201 ENDING AT 36100,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER20;
CREATE TABLE CUSTOMER20
            (
            C_ID          INTEGER      NOT NULL,
            C_STATE       CHAR(2)      NOT NULL,
            C_ZIP         CHAR(9)      NOT NULL,
            C_PHONE       CHAR(16)     NOT NULL,
            C_SINCE       BIGINT       NOT NULL,
            C_CREDIT_LIM  BIGINT       NOT NULL,
            C_MIDDLE      CHAR(2)      NOT NULL,
            C_CREDIT      CHAR(2)      NOT NULL,
            C_DISCOUNT    INTEGER      NOT NULL,
            C_DATA        VARCHAR(500) NOT NULL,
            C_LAST        VARCHAR(16)  NOT NULL,
            C_FIRST       VARCHAR(16)  NOT NULL,
            C_STREET_1    VARCHAR(20)  NOT NULL,
            C_STREET_2    VARCHAR(20)  NOT NULL,
            C_CITY        VARCHAR(20)  NOT NULL,
            C_D_ID        SMALLINT     NOT NULL,
            C_W_ID        INTEGER      NOT NULL,
            C_DELIVERY_CNT INTEGER     NOT NULL,
            C_BALANCE     BIGINT       NOT NULL,
            C_YTD_PAYMENT  BIGINT      NOT NULL,
            C_PAYMENT_CNT  INTEGER     NOT NULL
            )
            IN ts_customer_20
            INDEX IN is_customer_20
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 36101 ENDING AT 38000,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER21;
CREATE TABLE CUSTOMER21
            (
            C_ID          INTEGER      NOT NULL,
            C_STATE       CHAR(2)      NOT NULL,
            C_ZIP         CHAR(9)      NOT NULL,
            C_PHONE       CHAR(16)     NOT NULL,
            C_SINCE       BIGINT       NOT NULL,
            C_CREDIT_LIM  BIGINT       NOT NULL,
            C_MIDDLE      CHAR(2)      NOT NULL,
            C_CREDIT      CHAR(2)      NOT NULL,
```

```
            C_DISCOUNT     INTEGER       NOT NULL,
            C_DATA         VARCHAR(500)  NOT NULL,
            C_LAST         VARCHAR(16)   NOT NULL,
            C_FIRST        VARCHAR(16)   NOT NULL,
            C_STREET_1     VARCHAR(20)   NOT NULL,
            C_STREET_2     VARCHAR(20)   NOT NULL,
            C_CITY         VARCHAR(20)   NOT NULL,
            C_D_ID         SMALLINT      NOT NULL,
            C_W_ID         INTEGER       NOT NULL,
            C_DELIVERY_CNT INTEGER       NOT NULL,
            C_BALANCE      BIGINT        NOT NULL,
            C_YTD_PAYMENT  BIGINT        NOT NULL,
            C_PAYMENT_CNT  INTEGER       NOT NULL
            )
            IN ts_customer_21
            INDEX IN is_customer_21
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 38001 ENDING AT 39900,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER22;
CREATE TABLE CUSTOMER22
            (
            C_ID          INTEGER       NOT NULL,
            C_STATE       CHAR(2)       NOT NULL,
            C_ZIP         CHAR(9)       NOT NULL,
            C_PHONE       CHAR(16)      NOT NULL,
            C_SINCE       BIGINT        NOT NULL,
            C_CREDIT_LIM  BIGINT        NOT NULL,
            C_MIDDLE      CHAR(2)       NOT NULL,
            C_CREDIT      CHAR(2)       NOT NULL,
            C_DISCOUNT    INTEGER       NOT NULL,
            C_DATA        VARCHAR(500)  NOT NULL,
            C_LAST        VARCHAR(16)   NOT NULL,
            C_FIRST       VARCHAR(16)   NOT NULL,
            C_STREET_1    VARCHAR(20)   NOT NULL,
            C_STREET_2    VARCHAR(20)   NOT NULL,
            C_CITY        VARCHAR(20)   NOT NULL,
            C_D_ID        SMALLINT      NOT NULL,
            C_W_ID        INTEGER       NOT NULL,
            C_DELIVERY_CNT INTEGER      NOT NULL,
            C_BALANCE     BIGINT        NOT NULL,
            C_YTD_PAYMENT  BIGINT       NOT NULL,
            C_PAYMENT_CNT  INTEGER      NOT NULL
            )
            IN ts_customer_22
            INDEX IN is_customer_22
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 39901 ENDING AT 41800,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER23;
CREATE TABLE CUSTOMER23
            (
            C_ID          INTEGER       NOT NULL,
            C_STATE       CHAR(2)       NOT NULL,
            C_ZIP         CHAR(9)       NOT NULL,
            C_PHONE       CHAR(16)      NOT NULL,
            C_SINCE       BIGINT        NOT NULL,
            C_CREDIT_LIM  BIGINT        NOT NULL,
            C_MIDDLE      CHAR(2)       NOT NULL,
            C_CREDIT      CHAR(2)       NOT NULL,
            C_DISCOUNT    INTEGER       NOT NULL,
            C_DATA        VARCHAR(500)  NOT NULL,
            C_LAST        VARCHAR(16)   NOT NULL,
            C_FIRST       VARCHAR(16)   NOT NULL,
            C_STREET_1    VARCHAR(20)   NOT NULL,
            C_STREET_2    VARCHAR(20)   NOT NULL,
            C_CITY        VARCHAR(20)   NOT NULL,
            C_D_ID        SMALLINT      NOT NULL,
            C_W_ID        INTEGER       NOT NULL,
            C_DELIVERY_CNT INTEGER      NOT NULL,
            C_BALANCE     BIGINT        NOT NULL,
            C_YTD_PAYMENT  BIGINT       NOT NULL,
            C_PAYMENT_CNT  INTEGER      NOT NULL
            )
            IN ts_customer_23
            INDEX IN is_customer_23
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 41801 ENDING AT 43700,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER24;
CREATE TABLE CUSTOMER24
            (
            C_ID          INTEGER       NOT NULL,
            C_STATE       CHAR(2)       NOT NULL,
            C_ZIP         CHAR(9)       NOT NULL,
            C_PHONE       CHAR(16)      NOT NULL,
            C_SINCE       BIGINT        NOT NULL,
            C_CREDIT_LIM  BIGINT        NOT NULL,
            C_MIDDLE      CHAR(2)       NOT NULL,
            C_CREDIT      CHAR(2)       NOT NULL,
            C_DISCOUNT    INTEGER       NOT NULL,
            C_DATA        VARCHAR(500)  NOT NULL,
            C_LAST        VARCHAR(16)   NOT NULL,
            C_FIRST       VARCHAR(16)   NOT NULL,
            C_STREET_1    VARCHAR(20)   NOT NULL,
            C_STREET_2    VARCHAR(20)   NOT NULL,
            C_CITY        VARCHAR(20)   NOT NULL,
            C_D_ID        SMALLINT      NOT NULL,
            C_W_ID        INTEGER       NOT NULL,
            C_DELIVERY_CNT INTEGER      NOT NULL,
            C_BALANCE     BIGINT        NOT NULL,
            C_YTD_PAYMENT  BIGINT       NOT NULL,
            C_PAYMENT_CNT  INTEGER      NOT NULL
            )
            IN ts_customer_24
            INDEX IN is_customer_24
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 43701 ENDING AT 45600,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER25;
CREATE TABLE CUSTOMER25
            (
            C_ID          INTEGER       NOT NULL,
            C_STATE       CHAR(2)       NOT NULL,
            C_ZIP         CHAR(9)       NOT NULL,
            C_PHONE       CHAR(16)      NOT NULL,
            C_SINCE       BIGINT        NOT NULL,
            C_CREDIT_LIM  BIGINT        NOT NULL,
            C_MIDDLE      CHAR(2)       NOT NULL,
            C_CREDIT      CHAR(2)       NOT NULL,
            C_DISCOUNT    INTEGER       NOT NULL,
            C_DATA        VARCHAR(500)  NOT NULL,
            C_LAST        VARCHAR(16)   NOT NULL,
            C_FIRST       VARCHAR(16)   NOT NULL,
            C_STREET_1    VARCHAR(20)   NOT NULL,
            C_STREET_2    VARCHAR(20)   NOT NULL,
            C_CITY        VARCHAR(20)   NOT NULL,
            C_D_ID        SMALLINT      NOT NULL,
            C_W_ID        INTEGER       NOT NULL,
            C_DELIVERY_CNT INTEGER      NOT NULL,
            C_BALANCE     BIGINT        NOT NULL,
            C_YTD_PAYMENT  BIGINT       NOT NULL,
            C_PAYMENT_CNT  INTEGER      NOT NULL
            )
            IN ts_customer_25
            INDEX IN is_customer_25
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 45601 ENDING AT 47500,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER26;
CREATE TABLE CUSTOMER26
            (
            C_ID          INTEGER       NOT NULL,
            C_STATE       CHAR(2)       NOT NULL,
            C_ZIP         CHAR(9)       NOT NULL,
            C_PHONE       CHAR(16)      NOT NULL,
            C_SINCE       BIGINT        NOT NULL,
            C_CREDIT_LIM  BIGINT        NOT NULL,
            C_MIDDLE      CHAR(2)       NOT NULL,
            C_CREDIT      CHAR(2)       NOT NULL,
            C_DISCOUNT    INTEGER       NOT NULL,
            C_DATA        VARCHAR(500)  NOT NULL,
            C_LAST        VARCHAR(16)   NOT NULL,
            C_FIRST       VARCHAR(16)   NOT NULL,
            C_STREET_1    VARCHAR(20)   NOT NULL,
            C_STREET_2    VARCHAR(20)   NOT NULL,
            C_CITY        VARCHAR(20)   NOT NULL,
            C_D_ID        SMALLINT      NOT NULL,
            C_W_ID        INTEGER       NOT NULL,
            C_DELIVERY_CNT INTEGER      NOT NULL,
            C_BALANCE     BIGINT        NOT NULL,
            C_YTD_PAYMENT  BIGINT       NOT NULL,
            C_PAYMENT_CNT  INTEGER      NOT NULL
            )
            IN ts_customer_26
            INDEX IN is_customer_26
            ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 47501 ENDING AT 49400,
            C_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
```

```
connect to TPCC in share mode;
DROP TABLE CUSTOMER27;
CREATE TABLE CUSTOMER27
        (
         C_ID          INTEGER      NOT NULL,
         C_STATE       CHAR(2)      NOT NULL,
         C_ZIP         CHAR(9)      NOT NULL,
         C_PHONE       CHAR(16)     NOT NULL,
         C_SINCE       BIGINT       NOT NULL,
         C_CREDIT_LIM  BIGINT       NOT NULL,
         C_MIDDLE      CHAR(2)      NOT NULL,
         C_CREDIT      CHAR(2)      NOT NULL,
         C_DISCOUNT    INTEGER      NOT NULL,
         C_DATA        VARCHAR(500) NOT NULL,
         C_LAST        VARCHAR(16)  NOT NULL,
         C_FIRST       VARCHAR(16)  NOT NULL,
         C_STREET_1    VARCHAR(20)  NOT NULL,
         C_STREET_2    VARCHAR(20)  NOT NULL,
         C_CITY        VARCHAR(20)  NOT NULL,
         C_D_ID        SMALLINT     NOT NULL,
         C_W_ID        INTEGER      NOT NULL,
         C_DELIVERY_CNT INTEGER     NOT NULL,
         C_BALANCE     BIGINT       NOT NULL,
         C_YTD_PAYMENT BIGINT       NOT NULL,
         C_PAYMENT_CNT INTEGER      NOT NULL
        )
        IN ts_customer_27
        INDEX IN is_customer_27
        ORGANIZE BY KEY SEQUENCE (
         C_ID STARTING FROM 1 ENDING AT 3000,
         C_W_ID STARTING FROM 49401 ENDING AT 51300,
         C_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE CUSTOMER28;
CREATE TABLE CUSTOMER28
        (
         C_ID          INTEGER      NOT NULL,
         C_STATE       CHAR(2)      NOT NULL,
         C_ZIP         CHAR(9)      NOT NULL,
         C_PHONE       CHAR(16)     NOT NULL,
         C_SINCE       BIGINT       NOT NULL,
         C_CREDIT_LIM  BIGINT       NOT NULL,
         C_MIDDLE      CHAR(2)      NOT NULL,
         C_CREDIT      CHAR(2)      NOT NULL,
         C_DISCOUNT    INTEGER      NOT NULL,
         C_DATA        VARCHAR(500) NOT NULL,
         C_LAST        VARCHAR(16)  NOT NULL,
         C_FIRST       VARCHAR(16)  NOT NULL,
         C_STREET_1    VARCHAR(20)  NOT NULL,
         C_STREET_2    VARCHAR(20)  NOT NULL,
         C_CITY        VARCHAR(20)  NOT NULL,
         C_D_ID        SMALLINT     NOT NULL,
         C_W_ID        INTEGER      NOT NULL,
         C_DELIVERY_CNT INTEGER     NOT NULL,
         C_BALANCE     BIGINT       NOT NULL,
         C_YTD_PAYMENT BIGINT       NOT NULL,
         C_PAYMENT_CNT INTEGER      NOT NULL
        )
        IN ts_customer_28
        INDEX IN is_customer_28
        ORGANIZE BY KEY SEQUENCE (
         C_ID STARTING FROM 1 ENDING AT 3000,
         C_W_ID STARTING FROM 51301 ENDING AT 53200,
         C_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;

connect reset;
```

CRTB_DISTRICT.ddl

```
connect to TPCC in share mode;
DROP TABLE DISTRICT;
CREATE TABLE DISTRICT
        (
         D_NEXT_O_ID INTEGER      NOT NULL,
         D_TAX       INTEGER      NOT NULL,
         D_YTD       BIGINT       NOT NULL,
         D_NAME      CHAR(10)     NOT NULL,
         D_STREET_1  CHAR(20)     NOT NULL,
         D_STREET_2  CHAR(20)     NOT NULL,
         D_CITY      CHAR(20)     NOT NULL,
         D_STATE     CHAR(2)      NOT NULL,
         D_ZIP       CHAR(9)      NOT NULL,
         D_ID        SMALLINT     NOT NULL,
         D_W_ID      INTEGER      NOT NULL
        )
        IN ts_dist_01
        INDEX IN ts_dist_01
        ORGANIZE BY KEY SEQUENCE (
         D_ID STARTING FROM 1 ENDING AT 10,
         D_W_ID STARTING FROM 1 ENDING AT 53200
        )
        ALLOW OVERFLOW;

connect reset;
```

CRTB_HISTORY.ddl

```
connect to TPCC in share mode;
DROP TABLE HISTORY;
CREATE TABLE HISTORY
        (
         H_C_ID        INTEGER    NOT NULL,
         H_C_D_ID      SMALLINT   NOT NULL,
         H_C_W_ID      INTEGER    NOT NULL,
         H_D_ID        SMALLINT   NOT NULL,
         H_W_ID        INTEGER    NOT NULL,
         H_DATE        BIGINT     NOT NULL,
         H_AMOUNT      INTEGER    NOT NULL,
         H_DATA        CHAR(24)   NOT NULL
        )
        IN ts_history_01
        INDEX IN ts_history_01;
ALTER TABLE HISTORY APPEND ON;

connect reset;
```

CRTB_ITEM.ddl

```
connect to TPCC in share mode;
DROP TABLE ITEM;
CREATE TABLE ITEM
        (
         I_NAME        CHAR(24)   NOT NULL,
         I_PRICE       INTEGER    NOT NULL,
         I_DATA        VARCHAR(50) NOT NULL,
         I_IM_ID       INTEGER    NOT NULL,
         I_ID          INTEGER    NOT NULL
        )
        IN ts_item_01
        INDEX IN ts_item_01
        ORGANIZE BY KEY SEQUENCE (
         I_ID STARTING FROM 1 ENDING AT 100000
        )
        ALLOW OVERFLOW;
ALTER TABLE ITEM LOCKSIZE TABLE;

connect reset;
```

CRTB_NEW_ORDER.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDER;
CREATE TABLE NEW_ORDER
        (
         NO_O_ID       INTEGER    NOT NULL,
         NO_D_ID       SMALLINT   NOT NULL,
         NO_W_ID       INTEGER    NOT NULL
        )
        IN ts_neword_01
        INDEX IN is_neword_01;
ALTER TABLE NEW_ORDER APPEND ON;

connect reset;
```

CRTB_ORDERS.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDERS;
CREATE TABLE ORDERS
        (
         O_C_ID        INTEGER    NOT NULL,
         O_ENTRY_D     BIGINT     NOT NULL,
         O_CARRIER_ID  SMALLINT   NOT NULL,
         O_OL_CNT      SMALLINT   NOT NULL,
         O_ALL_LOCAL   SMALLINT   NOT NULL,
         O_ID          INTEGER    NOT NULL,
         O_W_ID        INTEGER    NOT NULL,
         O_D_ID        SMALLINT   NOT NULL
```

```
            )
            IN ts_order_01
            INDEX IN is_order_01
            ORGANIZE BY KEY SEQUENCE (
             O_ID STARTING FROM 1 ENDING AT 3750,
             O_W_ID STARTING FROM 1 ENDING AT 53200,
             O_D_ID STARTING FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;

connect reset;
```

CRTB_ORDER_LINE.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDER_LINE1;
CREATE TABLE ORDER_LINE1
            (
             OL_DELIVERY_D   BIGINT    NOT NULL,
             OL_AMOUNT       INTEGER   NOT NULL,
             OL_I_ID         INTEGER   NOT NULL,
             OL_SUPPLY_W_ID  INTEGER   NOT NULL,
             OL_QUANTITY     SMALLINT  NOT NULL,
             OL_DIST_INFO    CHAR(24)  NOT NULL,
             OL_O_ID         INTEGER   NOT NULL,
             OL_D_ID         SMALLINT  NOT NULL,
             OL_W_ID         INTEGER   NOT NULL,
             OL_NUMBER       SMALLINT  NOT NULL
            )
            IN ts_orderline_01
            INDEX IN ts_orderline_01
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 1 ENDING AT 1900,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3750,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE2;
CREATE TABLE ORDER_LINE2
            (
             OL_DELIVERY_D   BIGINT    NOT NULL,
             OL_AMOUNT       INTEGER   NOT NULL,
             OL_I_ID         INTEGER   NOT NULL,
             OL_SUPPLY_W_ID  INTEGER   NOT NULL,
             OL_QUANTITY     SMALLINT  NOT NULL,
             OL_DIST_INFO    CHAR(24)  NOT NULL,
             OL_O_ID         INTEGER   NOT NULL,
             OL_D_ID         SMALLINT  NOT NULL,
             OL_W_ID         INTEGER   NOT NULL,
             OL_NUMBER       SMALLINT  NOT NULL
            )
            IN ts_orderline_02
            INDEX IN ts_orderline_02
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 1901 ENDING AT 3800,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3750,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE3;
CREATE TABLE ORDER_LINE3
            (
             OL_DELIVERY_D   BIGINT    NOT NULL,
             OL_AMOUNT       INTEGER   NOT NULL,
             OL_I_ID         INTEGER   NOT NULL,
             OL_SUPPLY_W_ID  INTEGER   NOT NULL,
             OL_QUANTITY     SMALLINT  NOT NULL,
             OL_DIST_INFO    CHAR(24)  NOT NULL,
             OL_O_ID         INTEGER   NOT NULL,
             OL_D_ID         SMALLINT  NOT NULL,
             OL_W_ID         INTEGER   NOT NULL,
             OL_NUMBER       SMALLINT  NOT NULL
            )
            IN ts_orderline_03
            INDEX IN ts_orderline_03
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 3801 ENDING AT 5700,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3750,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE4;
CREATE TABLE ORDER_LINE4
            (
             OL_DELIVERY_D   BIGINT    NOT NULL,
             OL_AMOUNT       INTEGER   NOT NULL,
             OL_I_ID         INTEGER   NOT NULL,
             OL_SUPPLY_W_ID  INTEGER   NOT NULL,
             OL_QUANTITY     SMALLINT  NOT NULL,
             OL_DIST_INFO    CHAR(24)  NOT NULL,
             OL_O_ID         INTEGER   NOT NULL,
             OL_D_ID         SMALLINT  NOT NULL,
             OL_W_ID         INTEGER   NOT NULL,
             OL_NUMBER       SMALLINT  NOT NULL
            )
            IN ts_orderline_04
            INDEX IN ts_orderline_04
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 5701 ENDING AT 7600,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3750,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE5;
CREATE TABLE ORDER_LINE5
            (
             OL_DELIVERY_D   BIGINT    NOT NULL,
             OL_AMOUNT       INTEGER   NOT NULL,
             OL_I_ID         INTEGER   NOT NULL,
             OL_SUPPLY_W_ID  INTEGER   NOT NULL,
             OL_QUANTITY     SMALLINT  NOT NULL,
             OL_DIST_INFO    CHAR(24)  NOT NULL,
             OL_O_ID         INTEGER   NOT NULL,
             OL_D_ID         SMALLINT  NOT NULL,
             OL_W_ID         INTEGER   NOT NULL,
             OL_NUMBER       SMALLINT  NOT NULL
            )
            IN ts_orderline_05
            INDEX IN ts_orderline_05
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 7601 ENDING AT 9500,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3750,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE6;
CREATE TABLE ORDER_LINE6
            (
             OL_DELIVERY_D   BIGINT    NOT NULL,
             OL_AMOUNT       INTEGER   NOT NULL,
             OL_I_ID         INTEGER   NOT NULL,
             OL_SUPPLY_W_ID  INTEGER   NOT NULL,
             OL_QUANTITY     SMALLINT  NOT NULL,
             OL_DIST_INFO    CHAR(24)  NOT NULL,
             OL_O_ID         INTEGER   NOT NULL,
             OL_D_ID         SMALLINT  NOT NULL,
             OL_W_ID         INTEGER   NOT NULL,
             OL_NUMBER       SMALLINT  NOT NULL
            )
            IN ts_orderline_06
            INDEX IN ts_orderline_06
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 9501 ENDING AT 11400,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3750,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE7;
CREATE TABLE ORDER_LINE7
            (
             OL_DELIVERY_D   BIGINT    NOT NULL,
             OL_AMOUNT       INTEGER   NOT NULL,
             OL_I_ID         INTEGER   NOT NULL,
             OL_SUPPLY_W_ID  INTEGER   NOT NULL,
             OL_QUANTITY     SMALLINT  NOT NULL,
             OL_DIST_INFO    CHAR(24)  NOT NULL,
             OL_O_ID         INTEGER   NOT NULL,
             OL_D_ID         SMALLINT  NOT NULL,
             OL_W_ID         INTEGER   NOT NULL,
             OL_NUMBER       SMALLINT  NOT NULL
            )
            IN ts_orderline_07
            INDEX IN ts_orderline_07
            ORGANIZE BY KEY SEQUENCE (
             OL_W_ID STARTING FROM 11401 ENDING AT 13300,
             OL_D_ID STARTING FROM 1 ENDING AT 10,
             OL_O_ID STARTING FROM 1 ENDING AT 3750,
             OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE8;
CREATE TABLE ORDER_LINE8
            (
```

```
                OL_DELIVERY_D   BIGINT    NOT NULL,
                OL_AMOUNT       INTEGER   NOT NULL,
                OL_I_ID         INTEGER   NOT NULL,
                OL_SUPPLY_W_ID  INTEGER   NOT NULL,
                OL_QUANTITY     SMALLINT  NOT NULL,
                OL_DIST_INFO    CHAR(24)  NOT NULL,
                OL_O_ID         INTEGER   NOT NULL,
                OL_D_ID         SMALLINT  NOT NULL,
                OL_W_ID         INTEGER   NOT NULL,
                OL_NUMBER       SMALLINT  NOT NULL
               )
               IN ts_orderline_08
               INDEX IN ts_orderline_08
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 13301 ENDING AT 15200,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3750,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE9;
CREATE TABLE ORDER_LINE9
               (
                OL_DELIVERY_D   BIGINT    NOT NULL,
                OL_AMOUNT       INTEGER   NOT NULL,
                OL_I_ID         INTEGER   NOT NULL,
                OL_SUPPLY_W_ID  INTEGER   NOT NULL,
                OL_QUANTITY     SMALLINT  NOT NULL,
                OL_DIST_INFO    CHAR(24)  NOT NULL,
                OL_O_ID         INTEGER   NOT NULL,
                OL_D_ID         SMALLINT  NOT NULL,
                OL_W_ID         INTEGER   NOT NULL,
                OL_NUMBER       SMALLINT  NOT NULL
               )
               IN ts_orderline_09
               INDEX IN ts_orderline_09
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 15201 ENDING AT 17100,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3750,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE10;
CREATE TABLE ORDER_LINE10
               (
                OL_DELIVERY_D   BIGINT    NOT NULL,
                OL_AMOUNT       INTEGER   NOT NULL,
                OL_I_ID         INTEGER   NOT NULL,
                OL_SUPPLY_W_ID  INTEGER   NOT NULL,
                OL_QUANTITY     SMALLINT  NOT NULL,
                OL_DIST_INFO    CHAR(24)  NOT NULL,
                OL_O_ID         INTEGER   NOT NULL,
                OL_D_ID         SMALLINT  NOT NULL,
                OL_W_ID         INTEGER   NOT NULL,
                OL_NUMBER       SMALLINT  NOT NULL
               )
               IN ts_orderline_10
               INDEX IN ts_orderline_10
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 17101 ENDING AT 19000,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3750,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE11;
CREATE TABLE ORDER_LINE11
               (
                OL_DELIVERY_D   BIGINT    NOT NULL,
                OL_AMOUNT       INTEGER   NOT NULL,
                OL_I_ID         INTEGER   NOT NULL,
                OL_SUPPLY_W_ID  INTEGER   NOT NULL,
                OL_QUANTITY     SMALLINT  NOT NULL,
                OL_DIST_INFO    CHAR(24)  NOT NULL,
                OL_O_ID         INTEGER   NOT NULL,
                OL_D_ID         SMALLINT  NOT NULL,
                OL_W_ID         INTEGER   NOT NULL,
                OL_NUMBER       SMALLINT  NOT NULL
               )
               IN ts_orderline_11
               INDEX IN ts_orderline_11
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 19001 ENDING AT 20900,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3750,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE12;
CREATE TABLE ORDER_LINE12
               (
                OL_DELIVERY_D   BIGINT    NOT NULL,
                OL_AMOUNT       INTEGER   NOT NULL,
                OL_I_ID         INTEGER   NOT NULL,
                OL_SUPPLY_W_ID  INTEGER   NOT NULL,
                OL_QUANTITY     SMALLINT  NOT NULL,
                OL_DIST_INFO    CHAR(24)  NOT NULL,
                OL_O_ID         INTEGER   NOT NULL,
                OL_D_ID         SMALLINT  NOT NULL,
                OL_W_ID         INTEGER   NOT NULL,
                OL_NUMBER       SMALLINT  NOT NULL
               )
               IN ts_orderline_12
               INDEX IN ts_orderline_12
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 20901 ENDING AT 22800,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3750,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE13;
CREATE TABLE ORDER_LINE13
               (
                OL_DELIVERY_D   BIGINT    NOT NULL,
                OL_AMOUNT       INTEGER   NOT NULL,
                OL_I_ID         INTEGER   NOT NULL,
                OL_SUPPLY_W_ID  INTEGER   NOT NULL,
                OL_QUANTITY     SMALLINT  NOT NULL,
                OL_DIST_INFO    CHAR(24)  NOT NULL,
                OL_O_ID         INTEGER   NOT NULL,
                OL_D_ID         SMALLINT  NOT NULL,
                OL_W_ID         INTEGER   NOT NULL,
                OL_NUMBER       SMALLINT  NOT NULL
               )
               IN ts_orderline_13
               INDEX IN ts_orderline_13
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 22801 ENDING AT 24700,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3750,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE14;
CREATE TABLE ORDER_LINE14
               (
                OL_DELIVERY_D   BIGINT    NOT NULL,
                OL_AMOUNT       INTEGER   NOT NULL,
                OL_I_ID         INTEGER   NOT NULL,
                OL_SUPPLY_W_ID  INTEGER   NOT NULL,
                OL_QUANTITY     SMALLINT  NOT NULL,
                OL_DIST_INFO    CHAR(24)  NOT NULL,
                OL_O_ID         INTEGER   NOT NULL,
                OL_D_ID         SMALLINT  NOT NULL,
                OL_W_ID         INTEGER   NOT NULL,
                OL_NUMBER       SMALLINT  NOT NULL
               )
               IN ts_orderline_14
               INDEX IN ts_orderline_14
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 24701 ENDING AT 26600,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3750,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE15;
CREATE TABLE ORDER_LINE15
               (
                OL_DELIVERY_D   BIGINT    NOT NULL,
                OL_AMOUNT       INTEGER   NOT NULL,
                OL_I_ID         INTEGER   NOT NULL,
                OL_SUPPLY_W_ID  INTEGER   NOT NULL,
                OL_QUANTITY     SMALLINT  NOT NULL,
                OL_DIST_INFO    CHAR(24)  NOT NULL,
                OL_O_ID         INTEGER   NOT NULL,
                OL_D_ID         SMALLINT  NOT NULL,
                OL_W_ID         INTEGER   NOT NULL,
                OL_NUMBER       SMALLINT  NOT NULL
               )
               IN ts_orderline_15
               INDEX IN ts_orderline_15
               ORGANIZE BY KEY SEQUENCE (
                OL_W_ID STARTING FROM 26601 ENDING AT 28500,
                OL_D_ID STARTING FROM 1 ENDING AT 10,
                OL_O_ID STARTING FROM 1 ENDING AT 3750,
                OL_NUMBER STARTING FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE16;
```

```
CREATE TABLE ORDER_LINE16
        (
         OL_DELIVERY_D   BIGINT    NOT NULL,
         OL_AMOUNT       INTEGER   NOT NULL,
         OL_I_ID         INTEGER   NOT NULL,
         OL_SUPPLY_W_ID  INTEGER   NOT NULL,
         OL_QUANTITY     SMALLINT  NOT NULL,
         OL_DIST_INFO    CHAR(24)  NOT NULL,
         OL_O_ID         INTEGER   NOT NULL,
         OL_D_ID         SMALLINT  NOT NULL,
         OL_W_ID         INTEGER   NOT NULL,
         OL_NUMBER       SMALLINT  NOT NULL
        )
        IN ts_orderline_16
        INDEX IN ts_orderline_16
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 28501 ENDING AT 30400,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3750,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE17;
CREATE TABLE ORDER_LINE17
        (
         OL_DELIVERY_D   BIGINT    NOT NULL,
         OL_AMOUNT       INTEGER   NOT NULL,
         OL_I_ID         INTEGER   NOT NULL,
         OL_SUPPLY_W_ID  INTEGER   NOT NULL,
         OL_QUANTITY     SMALLINT  NOT NULL,
         OL_DIST_INFO    CHAR(24)  NOT NULL,
         OL_O_ID         INTEGER   NOT NULL,
         OL_D_ID         SMALLINT  NOT NULL,
         OL_W_ID         INTEGER   NOT NULL,
         OL_NUMBER       SMALLINT  NOT NULL
        )
        IN ts_orderline_17
        INDEX IN ts_orderline_17
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 30401 ENDING AT 32300,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3750,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE18;
CREATE TABLE ORDER_LINE18
        (
         OL_DELIVERY_D   BIGINT    NOT NULL,
         OL_AMOUNT       INTEGER   NOT NULL,
         OL_I_ID         INTEGER   NOT NULL,
         OL_SUPPLY_W_ID  INTEGER   NOT NULL,
         OL_QUANTITY     SMALLINT  NOT NULL,
         OL_DIST_INFO    CHAR(24)  NOT NULL,
         OL_O_ID         INTEGER   NOT NULL,
         OL_D_ID         SMALLINT  NOT NULL,
         OL_W_ID         INTEGER   NOT NULL,
         OL_NUMBER       SMALLINT  NOT NULL
        )
        IN ts_orderline_18
        INDEX IN ts_orderline_18
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 32301 ENDING AT 34200,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3750,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE19;
CREATE TABLE ORDER_LINE19
        (
         OL_DELIVERY_D   BIGINT    NOT NULL,
         OL_AMOUNT       INTEGER   NOT NULL,
         OL_I_ID         INTEGER   NOT NULL,
         OL_SUPPLY_W_ID  INTEGER   NOT NULL,
         OL_QUANTITY     SMALLINT  NOT NULL,
         OL_DIST_INFO    CHAR(24)  NOT NULL,
         OL_O_ID         INTEGER   NOT NULL,
         OL_D_ID         SMALLINT  NOT NULL,
         OL_W_ID         INTEGER   NOT NULL,
         OL_NUMBER       SMALLINT  NOT NULL
        )
        IN ts_orderline_19
        INDEX IN ts_orderline_19
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 34201 ENDING AT 36100,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3750,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE20;
CREATE TABLE ORDER_LINE20
        (
         OL_DELIVERY_D   BIGINT    NOT NULL,
         OL_AMOUNT       INTEGER   NOT NULL,
         OL_I_ID         INTEGER   NOT NULL,
         OL_SUPPLY_W_ID  INTEGER   NOT NULL,
         OL_QUANTITY     SMALLINT  NOT NULL,
         OL_DIST_INFO    CHAR(24)  NOT NULL,
         OL_O_ID         INTEGER   NOT NULL,
         OL_D_ID         SMALLINT  NOT NULL,
         OL_W_ID         INTEGER   NOT NULL,
         OL_NUMBER       SMALLINT  NOT NULL
        )
        IN ts_orderline_20
        INDEX IN ts_orderline_20
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 36101 ENDING AT 38000,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3750,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE21;
CREATE TABLE ORDER_LINE21
        (
         OL_DELIVERY_D   BIGINT    NOT NULL,
         OL_AMOUNT       INTEGER   NOT NULL,
         OL_I_ID         INTEGER   NOT NULL,
         OL_SUPPLY_W_ID  INTEGER   NOT NULL,
         OL_QUANTITY     SMALLINT  NOT NULL,
         OL_DIST_INFO    CHAR(24)  NOT NULL,
         OL_O_ID         INTEGER   NOT NULL,
         OL_D_ID         SMALLINT  NOT NULL,
         OL_W_ID         INTEGER   NOT NULL,
         OL_NUMBER       SMALLINT  NOT NULL
        )
        IN ts_orderline_21
        INDEX IN ts_orderline_21
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 38001 ENDING AT 39900,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3750,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE22;
CREATE TABLE ORDER_LINE22
        (
         OL_DELIVERY_D   BIGINT    NOT NULL,
         OL_AMOUNT       INTEGER   NOT NULL,
         OL_I_ID         INTEGER   NOT NULL,
         OL_SUPPLY_W_ID  INTEGER   NOT NULL,
         OL_QUANTITY     SMALLINT  NOT NULL,
         OL_DIST_INFO    CHAR(24)  NOT NULL,
         OL_O_ID         INTEGER   NOT NULL,
         OL_D_ID         SMALLINT  NOT NULL,
         OL_W_ID         INTEGER   NOT NULL,
         OL_NUMBER       SMALLINT  NOT NULL
        )
        IN ts_orderline_22
        INDEX IN ts_orderline_22
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 39901 ENDING AT 41800,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3750,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE23;
CREATE TABLE ORDER_LINE23
        (
         OL_DELIVERY_D   BIGINT    NOT NULL,
         OL_AMOUNT       INTEGER   NOT NULL,
         OL_I_ID         INTEGER   NOT NULL,
         OL_SUPPLY_W_ID  INTEGER   NOT NULL,
         OL_QUANTITY     SMALLINT  NOT NULL,
         OL_DIST_INFO    CHAR(24)  NOT NULL,
         OL_O_ID         INTEGER   NOT NULL,
         OL_D_ID         SMALLINT  NOT NULL,
         OL_W_ID         INTEGER   NOT NULL,
         OL_NUMBER       SMALLINT  NOT NULL
        )
        IN ts_orderline_23
        INDEX IN ts_orderline_23
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 41801 ENDING AT 43700,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3750,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
```

```
connect to TPCC in share mode;
DROP TABLE ORDER_LINE24;
CREATE TABLE ORDER_LINE24
        (
         OL_DELIVERY_D   BIGINT    NOT NULL,
         OL_AMOUNT       INTEGER   NOT NULL,
         OL_I_ID         INTEGER   NOT NULL,
         OL_SUPPLY_W_ID  INTEGER   NOT NULL,
         OL_QUANTITY     SMALLINT  NOT NULL,
         OL_DIST_INFO    CHAR(24)  NOT NULL,
         OL_O_ID         INTEGER   NOT NULL,
         OL_D_ID         SMALLINT  NOT NULL,
         OL_W_ID         INTEGER   NOT NULL,
         OL_NUMBER       SMALLINT  NOT NULL
        )
        IN ts_orderline_24
        INDEX IN ts_orderline_24
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 43701 ENDING AT 45600,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3750,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE25;
CREATE TABLE ORDER_LINE25
        (
         OL_DELIVERY_D   BIGINT    NOT NULL,
         OL_AMOUNT       INTEGER   NOT NULL,
         OL_I_ID         INTEGER   NOT NULL,
         OL_SUPPLY_W_ID  INTEGER   NOT NULL,
         OL_QUANTITY     SMALLINT  NOT NULL,
         OL_DIST_INFO    CHAR(24)  NOT NULL,
         OL_O_ID         INTEGER   NOT NULL,
         OL_D_ID         SMALLINT  NOT NULL,
         OL_W_ID         INTEGER   NOT NULL,
         OL_NUMBER       SMALLINT  NOT NULL
        )
        IN ts_orderline_25
        INDEX IN ts_orderline_25
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 45601 ENDING AT 47500,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3750,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE26;
CREATE TABLE ORDER_LINE26
        (
         OL_DELIVERY_D   BIGINT    NOT NULL,
         OL_AMOUNT       INTEGER   NOT NULL,
         OL_I_ID         INTEGER   NOT NULL,
         OL_SUPPLY_W_ID  INTEGER   NOT NULL,
         OL_QUANTITY     SMALLINT  NOT NULL,
         OL_DIST_INFO    CHAR(24)  NOT NULL,
         OL_O_ID         INTEGER   NOT NULL,
         OL_D_ID         SMALLINT  NOT NULL,
         OL_W_ID         INTEGER   NOT NULL,
         OL_NUMBER       SMALLINT  NOT NULL
        )
        IN ts_orderline_26
        INDEX IN ts_orderline_26
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 47501 ENDING AT 49400,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3750,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE27;
CREATE TABLE ORDER_LINE27
        (
         OL_DELIVERY_D   BIGINT    NOT NULL,
         OL_AMOUNT       INTEGER   NOT NULL,
         OL_I_ID         INTEGER   NOT NULL,
         OL_SUPPLY_W_ID  INTEGER   NOT NULL,
         OL_QUANTITY     SMALLINT  NOT NULL,
         OL_DIST_INFO    CHAR(24)  NOT NULL,
         OL_O_ID         INTEGER   NOT NULL,
         OL_D_ID         SMALLINT  NOT NULL,
         OL_W_ID         INTEGER   NOT NULL,
         OL_NUMBER       SMALLINT  NOT NULL
        )
        IN ts_orderline_27
        INDEX IN ts_orderline_27
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 49401 ENDING AT 51300,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3750,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDER_LINE28;
CREATE TABLE ORDER_LINE28
        (
         OL_DELIVERY_D   BIGINT    NOT NULL,
         OL_AMOUNT       INTEGER   NOT NULL,
         OL_I_ID         INTEGER   NOT NULL,
         OL_SUPPLY_W_ID  INTEGER   NOT NULL,
         OL_QUANTITY     SMALLINT  NOT NULL,
         OL_DIST_INFO    CHAR(24)  NOT NULL,
         OL_O_ID         INTEGER   NOT NULL,
         OL_D_ID         SMALLINT  NOT NULL,
         OL_W_ID         INTEGER   NOT NULL,
         OL_NUMBER       SMALLINT  NOT NULL
        )
        IN ts_orderline_28
        INDEX IN ts_orderline_28
        ORGANIZE BY KEY SEQUENCE (
         OL_W_ID STARTING FROM 51301 ENDING AT 53200,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3750,
         OL_NUMBER STARTING FROM 1 ENDING AT 15
        )
        ALLOW OVERFLOW;

connect reset;
```

CRTB_STOCK.DDL

```
connect to TPCC in share mode;
DROP TABLE STOCK1;
CREATE TABLE STOCK1
        (
         S_REMOTE_CNT    INTEGER     NOT NULL,
         S_QUANTITY      INTEGER     NOT NULL,
         S_ORDER_CNT     INTEGER     NOT NULL,
         S_YTD           INTEGER     NOT NULL,
         S_DATA          VARCHAR(50) NOT NULL,
         S_DIST_01       CHAR(24)    NOT NULL,
         S_DIST_02       CHAR(24)    NOT NULL,
         S_DIST_03       CHAR(24)    NOT NULL,
         S_DIST_04       CHAR(24)    NOT NULL,
         S_DIST_05       CHAR(24)    NOT NULL,
         S_DIST_06       CHAR(24)    NOT NULL,
         S_DIST_07       CHAR(24)    NOT NULL,
         S_DIST_08       CHAR(24)    NOT NULL,
         S_DIST_09       CHAR(24)    NOT NULL,
         S_DIST_10       CHAR(24)    NOT NULL,
         S_I_ID          INTEGER     NOT NULL,
         S_W_ID          INTEGER     NOT NULL
        )
        IN ts_stock_01
        INDEX IN ts_stock_01
        ORGANIZE BY KEY SEQUENCE (
         S_I_ID STARTING FROM 1 ENDING AT 100000,
         S_W_ID STARTING FROM 1 ENDING AT 1900
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK2;
CREATE TABLE STOCK2
        (
         S_REMOTE_CNT    INTEGER     NOT NULL,
         S_QUANTITY      INTEGER     NOT NULL,
         S_ORDER_CNT     INTEGER     NOT NULL,
         S_YTD           INTEGER     NOT NULL,
         S_DATA          VARCHAR(50) NOT NULL,
         S_DIST_01       CHAR(24)    NOT NULL,
         S_DIST_02       CHAR(24)    NOT NULL,
         S_DIST_03       CHAR(24)    NOT NULL,
         S_DIST_04       CHAR(24)    NOT NULL,
         S_DIST_05       CHAR(24)    NOT NULL,
         S_DIST_06       CHAR(24)    NOT NULL,
         S_DIST_07       CHAR(24)    NOT NULL,
         S_DIST_08       CHAR(24)    NOT NULL,
         S_DIST_09       CHAR(24)    NOT NULL,
         S_DIST_10       CHAR(24)    NOT NULL,
         S_I_ID          INTEGER     NOT NULL,
         S_W_ID          INTEGER     NOT NULL
        )
        IN ts_stock_02
        INDEX IN ts_stock_02
        ORGANIZE BY KEY SEQUENCE (
         S_I_ID STARTING FROM 1 ENDING AT 100000,
         S_W_ID STARTING FROM 1901 ENDING AT 3800
        )
        ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK3;
CREATE TABLE STOCK3
        (
```

```
                S_REMOTE_CNT    INTEGER     NOT NULL,
                S_QUANTITY      INTEGER     NOT NULL,
                S_ORDER_CNT     INTEGER     NOT NULL,
                S_YTD           INTEGER     NOT NULL,
                S_DATA          VARCHAR(50) NOT NULL,
                S_DIST_01       CHAR(24)    NOT NULL,
                S_DIST_02       CHAR(24)    NOT NULL,
                S_DIST_03       CHAR(24)    NOT NULL,
                S_DIST_04       CHAR(24)    NOT NULL,
                S_DIST_05       CHAR(24)    NOT NULL,
                S_DIST_06       CHAR(24)    NOT NULL,
                S_DIST_07       CHAR(24)    NOT NULL,
                S_DIST_08       CHAR(24)    NOT NULL,
                S_DIST_09       CHAR(24)    NOT NULL,
                S_DIST_10       CHAR(24)    NOT NULL,
                S_I_ID          INTEGER     NOT NULL,
                S_W_ID          INTEGER     NOT NULL
                )
                IN ts_stock_03
                INDEX IN ts_stock_03
                ORGANIZE BY KEY SEQUENCE (
                 S_I_ID STARTING FROM 1 ENDING AT 100000,
                 S_W_ID STARTING FROM 3801 ENDING AT 5700
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK4;
CREATE TABLE STOCK4
                (
                 S_REMOTE_CNT    INTEGER     NOT NULL,
                 S_QUANTITY      INTEGER     NOT NULL,
                 S_ORDER_CNT     INTEGER     NOT NULL,
                 S_YTD           INTEGER     NOT NULL,
                 S_DATA          VARCHAR(50) NOT NULL,
                 S_DIST_01       CHAR(24)    NOT NULL,
                 S_DIST_02       CHAR(24)    NOT NULL,
                 S_DIST_03       CHAR(24)    NOT NULL,
                 S_DIST_04       CHAR(24)    NOT NULL,
                 S_DIST_05       CHAR(24)    NOT NULL,
                 S_DIST_06       CHAR(24)    NOT NULL,
                 S_DIST_07       CHAR(24)    NOT NULL,
                 S_DIST_08       CHAR(24)    NOT NULL,
                 S_DIST_09       CHAR(24)    NOT NULL,
                 S_DIST_10       CHAR(24)    NOT NULL,
                 S_I_ID          INTEGER     NOT NULL,
                 S_W_ID          INTEGER     NOT NULL
                )
                IN ts_stock_04
                INDEX IN ts_stock_04
                ORGANIZE BY KEY SEQUENCE (
                 S_I_ID STARTING FROM 1 ENDING AT 100000,
                 S_W_ID STARTING FROM 5701 ENDING AT 7600
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK5;
CREATE TABLE STOCK5
                (
                 S_REMOTE_CNT    INTEGER     NOT NULL,
                 S_QUANTITY      INTEGER     NOT NULL,
                 S_ORDER_CNT     INTEGER     NOT NULL,
                 S_YTD           INTEGER     NOT NULL,
                 S_DATA          VARCHAR(50) NOT NULL,
                 S_DIST_01       CHAR(24)    NOT NULL,
                 S_DIST_02       CHAR(24)    NOT NULL,
                 S_DIST_03       CHAR(24)    NOT NULL,
                 S_DIST_04       CHAR(24)    NOT NULL,
                 S_DIST_05       CHAR(24)    NOT NULL,
                 S_DIST_06       CHAR(24)    NOT NULL,
                 S_DIST_07       CHAR(24)    NOT NULL,
                 S_DIST_08       CHAR(24)    NOT NULL,
                 S_DIST_09       CHAR(24)    NOT NULL,
                 S_DIST_10       CHAR(24)    NOT NULL,
                 S_I_ID          INTEGER     NOT NULL,
                 S_W_ID          INTEGER     NOT NULL
                )
                IN ts_stock_05
                INDEX IN ts_stock_05
                ORGANIZE BY KEY SEQUENCE (
                 S_I_ID STARTING FROM 1 ENDING AT 100000,
                 S_W_ID STARTING FROM 7601 ENDING AT 9500
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK6;
CREATE TABLE STOCK6
                (
                 S_REMOTE_CNT    INTEGER     NOT NULL,
                 S_QUANTITY      INTEGER     NOT NULL,
                 S_ORDER_CNT     INTEGER     NOT NULL,
                 S_YTD           INTEGER     NOT NULL,
                 S_DATA          VARCHAR(50) NOT NULL,
                 S_DIST_01       CHAR(24)    NOT NULL,
                 S_DIST_02       CHAR(24)    NOT NULL,
                 S_DIST_03       CHAR(24)    NOT NULL,
                 S_DIST_04       CHAR(24)    NOT NULL,
                 S_DIST_05       CHAR(24)    NOT NULL,
                 S_DIST_06       CHAR(24)    NOT NULL,
                 S_DIST_07       CHAR(24)    NOT NULL,
                 S_DIST_08       CHAR(24)    NOT NULL,
                 S_DIST_09       CHAR(24)    NOT NULL,
                 S_DIST_10       CHAR(24)    NOT NULL,
                 S_I_ID          INTEGER     NOT NULL,
                 S_W_ID          INTEGER     NOT NULL
                )
                IN ts_stock_06
                INDEX IN ts_stock_06
                ORGANIZE BY KEY SEQUENCE (
                 S_I_ID STARTING FROM 1 ENDING AT 100000,
                 S_W_ID STARTING FROM 9501 ENDING AT 11400
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK7;
CREATE TABLE STOCK7
                (
                 S_REMOTE_CNT    INTEGER     NOT NULL,
                 S_QUANTITY      INTEGER     NOT NULL,
                 S_ORDER_CNT     INTEGER     NOT NULL,
                 S_YTD           INTEGER     NOT NULL,
                 S_DATA          VARCHAR(50) NOT NULL,
                 S_DIST_01       CHAR(24)    NOT NULL,
                 S_DIST_02       CHAR(24)    NOT NULL,
                 S_DIST_03       CHAR(24)    NOT NULL,
                 S_DIST_04       CHAR(24)    NOT NULL,
                 S_DIST_05       CHAR(24)    NOT NULL,
                 S_DIST_06       CHAR(24)    NOT NULL,
                 S_DIST_07       CHAR(24)    NOT NULL,
                 S_DIST_08       CHAR(24)    NOT NULL,
                 S_DIST_09       CHAR(24)    NOT NULL,
                 S_DIST_10       CHAR(24)    NOT NULL,
                 S_I_ID          INTEGER     NOT NULL,
                 S_W_ID          INTEGER     NOT NULL
                )
                IN ts_stock_07
                INDEX IN ts_stock_07
                ORGANIZE BY KEY SEQUENCE (
                 S_I_ID STARTING FROM 1 ENDING AT 100000,
                 S_W_ID STARTING FROM 11401 ENDING AT 13300
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK8;
CREATE TABLE STOCK8
                (
                 S_REMOTE_CNT    INTEGER     NOT NULL,
                 S_QUANTITY      INTEGER     NOT NULL,
                 S_ORDER_CNT     INTEGER     NOT NULL,
                 S_YTD           INTEGER     NOT NULL,
                 S_DATA          VARCHAR(50) NOT NULL,
                 S_DIST_01       CHAR(24)    NOT NULL,
                 S_DIST_02       CHAR(24)    NOT NULL,
                 S_DIST_03       CHAR(24)    NOT NULL,
                 S_DIST_04       CHAR(24)    NOT NULL,
                 S_DIST_05       CHAR(24)    NOT NULL,
                 S_DIST_06       CHAR(24)    NOT NULL,
                 S_DIST_07       CHAR(24)    NOT NULL,
                 S_DIST_08       CHAR(24)    NOT NULL,
                 S_DIST_09       CHAR(24)    NOT NULL,
                 S_DIST_10       CHAR(24)    NOT NULL,
                 S_I_ID          INTEGER     NOT NULL,
                 S_W_ID          INTEGER     NOT NULL
                )
                IN ts_stock_08
                INDEX IN ts_stock_08
                ORGANIZE BY KEY SEQUENCE (
                 S_I_ID STARTING FROM 1 ENDING AT 100000,
                 S_W_ID STARTING FROM 13301 ENDING AT 15200
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK9;
CREATE TABLE STOCK9
                (
                 S_REMOTE_CNT    INTEGER     NOT NULL,
                 S_QUANTITY      INTEGER     NOT NULL,
                 S_ORDER_CNT     INTEGER     NOT NULL,
                 S_YTD           INTEGER     NOT NULL,
                 S_DATA          VARCHAR(50) NOT NULL,
                 S_DIST_01       CHAR(24)    NOT NULL,
                 S_DIST_02       CHAR(24)    NOT NULL,
                 S_DIST_03       CHAR(24)    NOT NULL,
                 S_DIST_04       CHAR(24)    NOT NULL,
                 S_DIST_05       CHAR(24)    NOT NULL,
                 S_DIST_06       CHAR(24)    NOT NULL,
                 S_DIST_07       CHAR(24)    NOT NULL,
                 S_DIST_08       CHAR(24)    NOT NULL,
                 S_DIST_09       CHAR(24)    NOT NULL,
                 S_DIST_10       CHAR(24)    NOT NULL,
                 S_I_ID          INTEGER     NOT NULL,
                 S_W_ID          INTEGER     NOT NULL
                )
```

```
            IN ts_stock_09
            INDEX IN ts_stock_09
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 15201 ENDING AT 17100
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK10;
CREATE TABLE STOCK10
            (
             S_REMOTE_CNT    INTEGER    NOT NULL,
             S_QUANTITY      INTEGER    NOT NULL,
             S_ORDER_CNT     INTEGER    NOT NULL,
             S_YTD           INTEGER    NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)   NOT NULL,
             S_DIST_02       CHAR(24)   NOT NULL,
             S_DIST_03       CHAR(24)   NOT NULL,
             S_DIST_04       CHAR(24)   NOT NULL,
             S_DIST_05       CHAR(24)   NOT NULL,
             S_DIST_06       CHAR(24)   NOT NULL,
             S_DIST_07       CHAR(24)   NOT NULL,
             S_DIST_08       CHAR(24)   NOT NULL,
             S_DIST_09       CHAR(24)   NOT NULL,
             S_DIST_10       CHAR(24)   NOT NULL,
             S_I_ID          INTEGER    NOT NULL,
             S_W_ID          INTEGER    NOT NULL
            )
            IN ts_stock_10
            INDEX IN ts_stock_10
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 17101 ENDING AT 19000
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK11;
CREATE TABLE STOCK11
            (
             S_REMOTE_CNT    INTEGER    NOT NULL,
             S_QUANTITY      INTEGER    NOT NULL,
             S_ORDER_CNT     INTEGER    NOT NULL,
             S_YTD           INTEGER    NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)   NOT NULL,
             S_DIST_02       CHAR(24)   NOT NULL,
             S_DIST_03       CHAR(24)   NOT NULL,
             S_DIST_04       CHAR(24)   NOT NULL,
             S_DIST_05       CHAR(24)   NOT NULL,
             S_DIST_06       CHAR(24)   NOT NULL,
             S_DIST_07       CHAR(24)   NOT NULL,
             S_DIST_08       CHAR(24)   NOT NULL,
             S_DIST_09       CHAR(24)   NOT NULL,
             S_DIST_10       CHAR(24)   NOT NULL,
             S_I_ID          INTEGER    NOT NULL,
             S_W_ID          INTEGER    NOT NULL
            )
            IN ts_stock_11
            INDEX IN ts_stock_11
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 19001 ENDING AT 20900
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK12;
CREATE TABLE STOCK12
            (
             S_REMOTE_CNT    INTEGER    NOT NULL,
             S_QUANTITY      INTEGER    NOT NULL,
             S_ORDER_CNT     INTEGER    NOT NULL,
             S_YTD           INTEGER    NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)   NOT NULL,
             S_DIST_02       CHAR(24)   NOT NULL,
             S_DIST_03       CHAR(24)   NOT NULL,
             S_DIST_04       CHAR(24)   NOT NULL,
             S_DIST_05       CHAR(24)   NOT NULL,
             S_DIST_06       CHAR(24)   NOT NULL,
             S_DIST_07       CHAR(24)   NOT NULL,
             S_DIST_08       CHAR(24)   NOT NULL,
             S_DIST_09       CHAR(24)   NOT NULL,
             S_DIST_10       CHAR(24)   NOT NULL,
             S_I_ID          INTEGER    NOT NULL,
             S_W_ID          INTEGER    NOT NULL
            )
            IN ts_stock_12
            INDEX IN ts_stock_12
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 20901 ENDING AT 22800
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK13;
CREATE TABLE STOCK13
            (
             S_REMOTE_CNT    INTEGER    NOT NULL,
             S_QUANTITY      INTEGER    NOT NULL,
             S_ORDER_CNT     INTEGER    NOT NULL,
             S_YTD           INTEGER    NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)   NOT NULL,
             S_DIST_02       CHAR(24)   NOT NULL,
             S_DIST_03       CHAR(24)   NOT NULL,
             S_DIST_04       CHAR(24)   NOT NULL,
             S_DIST_05       CHAR(24)   NOT NULL,
             S_DIST_06       CHAR(24)   NOT NULL,
             S_DIST_07       CHAR(24)   NOT NULL,
             S_DIST_08       CHAR(24)   NOT NULL,
             S_DIST_09       CHAR(24)   NOT NULL,
             S_DIST_10       CHAR(24)   NOT NULL,
             S_I_ID          INTEGER    NOT NULL,
             S_W_ID          INTEGER    NOT NULL
            )
            IN ts_stock_13
            INDEX IN ts_stock_13
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 22801 ENDING AT 24700
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK14;
CREATE TABLE STOCK14
            (
             S_REMOTE_CNT    INTEGER    NOT NULL,
             S_QUANTITY      INTEGER    NOT NULL,
             S_ORDER_CNT     INTEGER    NOT NULL,
             S_YTD           INTEGER    NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)   NOT NULL,
             S_DIST_02       CHAR(24)   NOT NULL,
             S_DIST_03       CHAR(24)   NOT NULL,
             S_DIST_04       CHAR(24)   NOT NULL,
             S_DIST_05       CHAR(24)   NOT NULL,
             S_DIST_06       CHAR(24)   NOT NULL,
             S_DIST_07       CHAR(24)   NOT NULL,
             S_DIST_08       CHAR(24)   NOT NULL,
             S_DIST_09       CHAR(24)   NOT NULL,
             S_DIST_10       CHAR(24)   NOT NULL,
             S_I_ID          INTEGER    NOT NULL,
             S_W_ID          INTEGER    NOT NULL
            )
            IN ts_stock_14
            INDEX IN ts_stock_14
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 24701 ENDING AT 26600
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK15;
CREATE TABLE STOCK15
            (
             S_REMOTE_CNT    INTEGER    NOT NULL,
             S_QUANTITY      INTEGER    NOT NULL,
             S_ORDER_CNT     INTEGER    NOT NULL,
             S_YTD           INTEGER    NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)   NOT NULL,
             S_DIST_02       CHAR(24)   NOT NULL,
             S_DIST_03       CHAR(24)   NOT NULL,
             S_DIST_04       CHAR(24)   NOT NULL,
             S_DIST_05       CHAR(24)   NOT NULL,
             S_DIST_06       CHAR(24)   NOT NULL,
             S_DIST_07       CHAR(24)   NOT NULL,
             S_DIST_08       CHAR(24)   NOT NULL,
             S_DIST_09       CHAR(24)   NOT NULL,
             S_DIST_10       CHAR(24)   NOT NULL,
             S_I_ID          INTEGER    NOT NULL,
             S_W_ID          INTEGER    NOT NULL
            )
            IN ts_stock_15
            INDEX IN ts_stock_15
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 26601 ENDING AT 28500
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK16;
CREATE TABLE STOCK16
            (
             S_REMOTE_CNT    INTEGER    NOT NULL,
             S_QUANTITY      INTEGER    NOT NULL,
             S_ORDER_CNT     INTEGER    NOT NULL,
             S_YTD           INTEGER    NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)   NOT NULL,
```

```
                S_DIST_02       CHAR(24)    NOT NULL,
                S_DIST_03       CHAR(24)    NOT NULL,
                S_DIST_04       CHAR(24)    NOT NULL,
                S_DIST_05       CHAR(24)    NOT NULL,
                S_DIST_06       CHAR(24)    NOT NULL,
                S_DIST_07       CHAR(24)    NOT NULL,
                S_DIST_08       CHAR(24)    NOT NULL,
                S_DIST_09       CHAR(24)    NOT NULL,
                S_DIST_10       CHAR(24)    NOT NULL,
                S_I_ID          INTEGER     NOT NULL,
                S_W_ID          INTEGER     NOT NULL
                )
                IN ts_stock_16
                INDEX IN ts_stock_16
                ORGANIZE BY KEY SEQUENCE (
                 S_I_ID STARTING FROM 1 ENDING AT 100000,
                 S_W_ID STARTING FROM 28501 ENDING AT 30400
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK17;
CREATE TABLE STOCK17
                (
                 S_REMOTE_CNT    INTEGER     NOT NULL,
                 S_QUANTITY      INTEGER     NOT NULL,
                 S_ORDER_CNT     INTEGER     NOT NULL,
                 S_YTD           INTEGER     NOT NULL,
                 S_DATA          VARCHAR(50) NOT NULL,
                 S_DIST_01       CHAR(24)    NOT NULL,
                 S_DIST_02       CHAR(24)    NOT NULL,
                 S_DIST_03       CHAR(24)    NOT NULL,
                 S_DIST_04       CHAR(24)    NOT NULL,
                 S_DIST_05       CHAR(24)    NOT NULL,
                 S_DIST_06       CHAR(24)    NOT NULL,
                 S_DIST_07       CHAR(24)    NOT NULL,
                 S_DIST_08       CHAR(24)    NOT NULL,
                 S_DIST_09       CHAR(24)    NOT NULL,
                 S_DIST_10       CHAR(24)    NOT NULL,
                 S_I_ID          INTEGER     NOT NULL,
                 S_W_ID          INTEGER     NOT NULL
                )
                IN ts_stock_17
                INDEX IN ts_stock_17
                ORGANIZE BY KEY SEQUENCE (
                 S_I_ID STARTING FROM 1 ENDING AT 100000,
                 S_W_ID STARTING FROM 30401 ENDING AT 32300
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK18;
CREATE TABLE STOCK18
                (
                 S_REMOTE_CNT    INTEGER     NOT NULL,
                 S_QUANTITY      INTEGER     NOT NULL,
                 S_ORDER_CNT     INTEGER     NOT NULL,
                 S_YTD           INTEGER     NOT NULL,
                 S_DATA          VARCHAR(50) NOT NULL,
                 S_DIST_01       CHAR(24)    NOT NULL,
                 S_DIST_02       CHAR(24)    NOT NULL,
                 S_DIST_03       CHAR(24)    NOT NULL,
                 S_DIST_04       CHAR(24)    NOT NULL,
                 S_DIST_05       CHAR(24)    NOT NULL,
                 S_DIST_06       CHAR(24)    NOT NULL,
                 S_DIST_07       CHAR(24)    NOT NULL,
                 S_DIST_08       CHAR(24)    NOT NULL,
                 S_DIST_09       CHAR(24)    NOT NULL,
                 S_DIST_10       CHAR(24)    NOT NULL,
                 S_I_ID          INTEGER     NOT NULL,
                 S_W_ID          INTEGER     NOT NULL
                )
                IN ts_stock_18
                INDEX IN ts_stock_18
                ORGANIZE BY KEY SEQUENCE (
                 S_I_ID STARTING FROM 1 ENDING AT 100000,
                 S_W_ID STARTING FROM 32301 ENDING AT 34200
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK19;
CREATE TABLE STOCK19
                (
                 S_REMOTE_CNT    INTEGER     NOT NULL,
                 S_QUANTITY      INTEGER     NOT NULL,
                 S_ORDER_CNT     INTEGER     NOT NULL,
                 S_YTD           INTEGER     NOT NULL,
                 S_DATA          VARCHAR(50) NOT NULL,
                 S_DIST_01       CHAR(24)    NOT NULL,
                 S_DIST_02       CHAR(24)    NOT NULL,
                 S_DIST_03       CHAR(24)    NOT NULL,
                 S_DIST_04       CHAR(24)    NOT NULL,
                 S_DIST_05       CHAR(24)    NOT NULL,
                 S_DIST_06       CHAR(24)    NOT NULL,
                 S_DIST_07       CHAR(24)    NOT NULL,
                 S_DIST_08       CHAR(24)    NOT NULL,
                 S_DIST_09       CHAR(24)    NOT NULL,
                 S_DIST_10       CHAR(24)    NOT NULL,
                 S_I_ID          INTEGER     NOT NULL,
                 S_W_ID          INTEGER     NOT NULL
                )
                IN ts_stock_19
                INDEX IN ts_stock_19
                ORGANIZE BY KEY SEQUENCE (
                 S_I_ID STARTING FROM 1 ENDING AT 100000,
                 S_W_ID STARTING FROM 34201 ENDING AT 36100
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK20;
CREATE TABLE STOCK20
                (
                 S_REMOTE_CNT    INTEGER     NOT NULL,
                 S_QUANTITY      INTEGER     NOT NULL,
                 S_ORDER_CNT     INTEGER     NOT NULL,
                 S_YTD           INTEGER     NOT NULL,
                 S_DATA          VARCHAR(50) NOT NULL,
                 S_DIST_01       CHAR(24)    NOT NULL,
                 S_DIST_02       CHAR(24)    NOT NULL,
                 S_DIST_03       CHAR(24)    NOT NULL,
                 S_DIST_04       CHAR(24)    NOT NULL,
                 S_DIST_05       CHAR(24)    NOT NULL,
                 S_DIST_06       CHAR(24)    NOT NULL,
                 S_DIST_07       CHAR(24)    NOT NULL,
                 S_DIST_08       CHAR(24)    NOT NULL,
                 S_DIST_09       CHAR(24)    NOT NULL,
                 S_DIST_10       CHAR(24)    NOT NULL,
                 S_I_ID          INTEGER     NOT NULL,
                 S_W_ID          INTEGER     NOT NULL
                )
                IN ts_stock_20
                INDEX IN ts_stock_20
                ORGANIZE BY KEY SEQUENCE (
                 S_I_ID STARTING FROM 1 ENDING AT 100000,
                 S_W_ID STARTING FROM 36101 ENDING AT 38000
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK21;
CREATE TABLE STOCK21
                (
                 S_REMOTE_CNT    INTEGER     NOT NULL,
                 S_QUANTITY      INTEGER     NOT NULL,
                 S_ORDER_CNT     INTEGER     NOT NULL,
                 S_YTD           INTEGER     NOT NULL,
                 S_DATA          VARCHAR(50) NOT NULL,
                 S_DIST_01       CHAR(24)    NOT NULL,
                 S_DIST_02       CHAR(24)    NOT NULL,
                 S_DIST_03       CHAR(24)    NOT NULL,
                 S_DIST_04       CHAR(24)    NOT NULL,
                 S_DIST_05       CHAR(24)    NOT NULL,
                 S_DIST_06       CHAR(24)    NOT NULL,
                 S_DIST_07       CHAR(24)    NOT NULL,
                 S_DIST_08       CHAR(24)    NOT NULL,
                 S_DIST_09       CHAR(24)    NOT NULL,
                 S_DIST_10       CHAR(24)    NOT NULL,
                 S_I_ID          INTEGER     NOT NULL,
                 S_W_ID          INTEGER     NOT NULL
                )
                IN ts_stock_21
                INDEX IN ts_stock_21
                ORGANIZE BY KEY SEQUENCE (
                 S_I_ID STARTING FROM 1 ENDING AT 100000,
                 S_W_ID STARTING FROM 38001 ENDING AT 39900
                )
                ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK22;
CREATE TABLE STOCK22
                (
                 S_REMOTE_CNT    INTEGER     NOT NULL,
                 S_QUANTITY      INTEGER     NOT NULL,
                 S_ORDER_CNT     INTEGER     NOT NULL,
                 S_YTD           INTEGER     NOT NULL,
                 S_DATA          VARCHAR(50) NOT NULL,
                 S_DIST_01       CHAR(24)    NOT NULL,
                 S_DIST_02       CHAR(24)    NOT NULL,
                 S_DIST_03       CHAR(24)    NOT NULL,
                 S_DIST_04       CHAR(24)    NOT NULL,
                 S_DIST_05       CHAR(24)    NOT NULL,
                 S_DIST_06       CHAR(24)    NOT NULL,
                 S_DIST_07       CHAR(24)    NOT NULL,
                 S_DIST_08       CHAR(24)    NOT NULL,
                 S_DIST_09       CHAR(24)    NOT NULL,
                 S_DIST_10       CHAR(24)    NOT NULL,
                 S_I_ID          INTEGER     NOT NULL,
                 S_W_ID          INTEGER     NOT NULL
                )
                IN ts_stock_22
                INDEX IN ts_stock_22
                ORGANIZE BY KEY SEQUENCE (
                 S_I_ID STARTING FROM 1 ENDING AT 100000,
                 S_W_ID STARTING FROM 39901 ENDING AT 41800
                )
```

```
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK23;
CREATE TABLE STOCK23
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_23
            INDEX IN ts_stock_23
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 41801 ENDING AT 43700
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK24;
CREATE TABLE STOCK24
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_24
            INDEX IN ts_stock_24
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 43701 ENDING AT 45600
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK25;
CREATE TABLE STOCK25
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_25
            INDEX IN ts_stock_25
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 45601 ENDING AT 47500
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK26;
CREATE TABLE STOCK26
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_26
            INDEX IN ts_stock_26
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 47501 ENDING AT 49400
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK27;
CREATE TABLE STOCK27
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_27
            INDEX IN ts_stock_27
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 49401 ENDING AT 51300
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE STOCK28;
CREATE TABLE STOCK28
            (
             S_REMOTE_CNT    INTEGER     NOT NULL,
             S_QUANTITY      INTEGER     NOT NULL,
             S_ORDER_CNT     INTEGER     NOT NULL,
             S_YTD           INTEGER     NOT NULL,
             S_DATA          VARCHAR(50) NOT NULL,
             S_DIST_01       CHAR(24)    NOT NULL,
             S_DIST_02       CHAR(24)    NOT NULL,
             S_DIST_03       CHAR(24)    NOT NULL,
             S_DIST_04       CHAR(24)    NOT NULL,
             S_DIST_05       CHAR(24)    NOT NULL,
             S_DIST_06       CHAR(24)    NOT NULL,
             S_DIST_07       CHAR(24)    NOT NULL,
             S_DIST_08       CHAR(24)    NOT NULL,
             S_DIST_09       CHAR(24)    NOT NULL,
             S_DIST_10       CHAR(24)    NOT NULL,
             S_I_ID          INTEGER     NOT NULL,
             S_W_ID          INTEGER     NOT NULL
            )
            IN ts_stock_28
            INDEX IN ts_stock_28
            ORGANIZE BY KEY SEQUENCE (
             S_I_ID STARTING FROM 1 ENDING AT 100000,
             S_W_ID STARTING FROM 51301 ENDING AT 53200
            )
            ALLOW OVERFLOW;

connect reset;
```

CRTB_WAREHOUSE.ddl

```
connect to TPCC in share mode;
DROP TABLE WAREHOUSE;
CREATE TABLE WAREHOUSE
            (
             W_NAME      CHAR(10)       NOT NULL,
             W_STREET_1  CHAR(20)       NOT NULL,
             W_STREET_2  CHAR(20)       NOT NULL,
```

```
            W_CITY      CHAR(20)       NOT NULL,
            W_STATE     CHAR(2)        NOT NULL,
            W_ZIP       CHAR(9)        NOT NULL,
            W_TAX       INTEGER        NOT NULL,
            W_YTD       BIGINT         NOT NULL,
            W_ID        INTEGER        NOT NULL
            )
            IN ts_ware_01
            INDEX IN ts_ware_01
            ORGANIZE BY KEY SEQUENCE (
            W_ID STARTING FROM 1 ENDING AT 53200
            )
            ALLOW OVERFLOW;

connect reset;
```

CRVW_CUSTOMER.ddl

```
connect to TPCC in share mode;
DROP VIEW CUSTOMER;
CREATE VIEW CUSTOMER
            (C_ID,
            C_STATE,
            C_ZIP,
            C_PHONE,
            C_SINCE,
            C_CREDIT_LIM,
            C_MIDDLE,
            C_CREDIT,
            C_DISCOUNT,
            C_DATA,
            C_LAST,
            C_FIRST,
            C_STREET_1,
            C_STREET_2,
            C_CITY,
            C_D_ID,
            C_W_ID,
            C_DELIVERY_CNT,
            C_BALANCE,
            C_YTD_PAYMENT,
            C_PAYMENT_CNT
            ) AS SELECT * FROM CUSTOMER1
            WHERE (C_W_ID BETWEEN 1 AND 1900) UNION ALL SELECT * FROM CUSTOMER2
            WHERE (C_W_ID BETWEEN 1901 AND 3800) UNION ALL SELECT * FROM
CUSTOMER3
            WHERE (C_W_ID BETWEEN 3801 AND 5700) UNION ALL SELECT * FROM
CUSTOMER4
            WHERE (C_W_ID BETWEEN 5701 AND 7600) UNION ALL SELECT * FROM
CUSTOMER5
            WHERE (C_W_ID BETWEEN 7601 AND 9500) UNION ALL SELECT * FROM
CUSTOMER6
            WHERE (C_W_ID BETWEEN 9501 AND 11400) UNION ALL SELECT * FROM
CUSTOMER7
            WHERE (C_W_ID BETWEEN 11401 AND 13300) UNION ALL SELECT * FROM
CUSTOMER8
            WHERE (C_W_ID BETWEEN 13301 AND 15200) UNION ALL SELECT * FROM
CUSTOMER9
            WHERE (C_W_ID BETWEEN 15201 AND 17100) UNION ALL SELECT * FROM
CUSTOMER10
            WHERE (C_W_ID BETWEEN 17101 AND 19000) UNION ALL SELECT * FROM
CUSTOMER11
            WHERE (C_W_ID BETWEEN 19001 AND 20900) UNION ALL SELECT * FROM
CUSTOMER12
            WHERE (C_W_ID BETWEEN 20901 AND 22800) UNION ALL SELECT * FROM
CUSTOMER13
            WHERE (C_W_ID BETWEEN 22801 AND 24700) UNION ALL SELECT * FROM
CUSTOMER14
            WHERE (C_W_ID BETWEEN 24701 AND 26600) UNION ALL SELECT * FROM
CUSTOMER15
            WHERE (C_W_ID BETWEEN 26601 AND 28500) UNION ALL SELECT * FROM
CUSTOMER16
            WHERE (C_W_ID BETWEEN 28501 AND 30400) UNION ALL SELECT * FROM
CUSTOMER17
            WHERE (C_W_ID BETWEEN 30401 AND 32300) UNION ALL SELECT * FROM
CUSTOMER18
            WHERE (C_W_ID BETWEEN 32301 AND 34200) UNION ALL SELECT * FROM
CUSTOMER19
            WHERE (C_W_ID BETWEEN 34201 AND 36100) UNION ALL SELECT * FROM
CUSTOMER20
            WHERE (C_W_ID BETWEEN 36101 AND 38000) UNION ALL SELECT * FROM
CUSTOMER21
            WHERE (C_W_ID BETWEEN 38001 AND 39900) UNION ALL SELECT * FROM
CUSTOMER22
            WHERE (C_W_ID BETWEEN 39901 AND 41800) UNION ALL SELECT * FROM
CUSTOMER23
            WHERE (C_W_ID BETWEEN 41801 AND 43700) UNION ALL SELECT * FROM
CUSTOMER24
            WHERE (C_W_ID BETWEEN 43701 AND 45600) UNION ALL SELECT * FROM
CUSTOMER25
            WHERE (C_W_ID BETWEEN 45601 AND 47500) UNION ALL SELECT * FROM
CUSTOMER26
            WHERE (C_W_ID BETWEEN 47501 AND 49400) UNION ALL SELECT * FROM
CUSTOMER27
            WHERE (C_W_ID BETWEEN 49401 AND 51300) UNION ALL SELECT * FROM
CUSTOMER28
            WHERE (C_W_ID >= 51301) WITH ROW MOVEMENT;
COMMIT WORK;

connect reset;
```

CRVW_ORDER_LINE.ddl

```
connect to TPCC in share mode;
DROP VIEW ORDER_LINE;
CREATE VIEW ORDER_LINE
            (OL_DELIVERY_D,
            OL_AMOUNT,
            OL_I_ID,
            OL_SUPPLY_W_ID,
            OL_QUANTITY,
            OL_DIST_INFO,
            OL_O_ID,
            OL_D_ID,
            OL_W_ID,
            OL_NUMBER
            ) AS SELECT * FROM ORDER_LINE1
            WHERE (OL_W_ID BETWEEN 1 AND 1900) UNION ALL SELECT * FROM
ORDER_LINE2
            WHERE (OL_W_ID BETWEEN 1901 AND 3800) UNION ALL SELECT * FROM
ORDER_LINE3
            WHERE (OL_W_ID BETWEEN 3801 AND 5700) UNION ALL SELECT * FROM
ORDER_LINE4
            WHERE (OL_W_ID BETWEEN 5701 AND 7600) UNION ALL SELECT * FROM
ORDER_LINE5
            WHERE (OL_W_ID BETWEEN 7601 AND 9500) UNION ALL SELECT * FROM
ORDER_LINE6
            WHERE (OL_W_ID BETWEEN 9501 AND 11400) UNION ALL SELECT * FROM
ORDER_LINE7
            WHERE (OL_W_ID BETWEEN 11401 AND 13300) UNION ALL SELECT * FROM
ORDER_LINE8
            WHERE (OL_W_ID BETWEEN 13301 AND 15200) UNION ALL SELECT * FROM
ORDER_LINE9
            WHERE (OL_W_ID BETWEEN 15201 AND 17100) UNION ALL SELECT * FROM
ORDER_LINE10
            WHERE (OL_W_ID BETWEEN 17101 AND 19000) UNION ALL SELECT * FROM
ORDER_LINE11
            WHERE (OL_W_ID BETWEEN 19001 AND 20900) UNION ALL SELECT * FROM
ORDER_LINE12
            WHERE (OL_W_ID BETWEEN 20901 AND 22800) UNION ALL SELECT * FROM
ORDER_LINE13
            WHERE (OL_W_ID BETWEEN 22801 AND 24700) UNION ALL SELECT * FROM
ORDER_LINE14
            WHERE (OL_W_ID BETWEEN 24701 AND 26600) UNION ALL SELECT * FROM
ORDER_LINE15
            WHERE (OL_W_ID BETWEEN 26601 AND 28500) UNION ALL SELECT * FROM
ORDER_LINE16
            WHERE (OL_W_ID BETWEEN 28501 AND 30400) UNION ALL SELECT * FROM
ORDER_LINE17
            WHERE (OL_W_ID BETWEEN 30401 AND 32300) UNION ALL SELECT * FROM
ORDER_LINE18
            WHERE (OL_W_ID BETWEEN 32301 AND 34200) UNION ALL SELECT * FROM
ORDER_LINE19
            WHERE (OL_W_ID BETWEEN 34201 AND 36100) UNION ALL SELECT * FROM
ORDER_LINE20
            WHERE (OL_W_ID BETWEEN 36101 AND 38000) UNION ALL SELECT * FROM
ORDER_LINE21
            WHERE (OL_W_ID BETWEEN 38001 AND 39900) UNION ALL SELECT * FROM
ORDER_LINE22
            WHERE (OL_W_ID BETWEEN 39901 AND 41800) UNION ALL SELECT * FROM
ORDER_LINE23
            WHERE (OL_W_ID BETWEEN 41801 AND 43700) UNION ALL SELECT * FROM
ORDER_LINE24
            WHERE (OL_W_ID BETWEEN 43701 AND 45600) UNION ALL SELECT * FROM
ORDER_LINE25
            WHERE (OL_W_ID BETWEEN 45601 AND 47500) UNION ALL SELECT * FROM
ORDER_LINE26
            WHERE (OL_W_ID BETWEEN 47501 AND 49400) UNION ALL SELECT * FROM
ORDER_LINE27
            WHERE (OL_W_ID BETWEEN 49401 AND 51300) UNION ALL SELECT * FROM
ORDER_LINE28
            WHERE (OL_W_ID >= 51301) WITH ROW MOVEMENT;
COMMIT WORK;

connect reset;
```

CRVW_STOCK.ddl

```
connect to TPCC in share mode;
DROP VIEW STOCK;
CREATE VIEW STOCK
            (S_REMOTE_CNT,
            S_QUANTITY,
            S_ORDER_CNT,
            S_YTD,
            S_DATA,
            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_I_ID,
            S_W_ID
            ) AS SELECT * FROM STOCK1
            WHERE (S_W_ID BETWEEN 1 AND 1900) UNION ALL SELECT * FROM STOCK2
            WHERE (S_W_ID BETWEEN 1901 AND 3800) UNION ALL SELECT * FROM STOCK3
            WHERE (S_W_ID BETWEEN 3801 AND 5700) UNION ALL SELECT * FROM STOCK4
            WHERE (S_W_ID BETWEEN 5701 AND 7600) UNION ALL SELECT * FROM STOCK5
            WHERE (S_W_ID BETWEEN 7601 AND 9500) UNION ALL SELECT * FROM STOCK6
            WHERE (S_W_ID BETWEEN 9501 AND 11400) UNION ALL SELECT * FROM STOCK7
            WHERE (S_W_ID BETWEEN 11401 AND 13300) UNION ALL SELECT * FROM STOCK8
            WHERE (S_W_ID BETWEEN 13301 AND 15200) UNION ALL SELECT * FROM STOCK9
            WHERE (S_W_ID BETWEEN 15201 AND 17100) UNION ALL SELECT * FROM
STOCK10
            WHERE (S_W_ID BETWEEN 17101 AND 19000) UNION ALL SELECT * FROM
STOCK11
            WHERE (S_W_ID BETWEEN 19001 AND 20900) UNION ALL SELECT * FROM
STOCK12
            WHERE (S_W_ID BETWEEN 20901 AND 22800) UNION ALL SELECT
* FROM STOCK13
            WHERE (S_W_ID BETWEEN 22801 AND 24700) UNION ALL SELECT * FROM
STOCK14
            WHERE (S_W_ID BETWEEN 24701 AND 26600) UNION ALL SELECT * FROM
STOCK15
            WHERE (S_W_ID BETWEEN 26601 AND 28500) UNION ALL SELECT * FROM
STOCK16
            WHERE (S_W_ID BETWEEN 28501 AND 30400) UNION ALL SELECT * FROM
STOCK17
            WHERE (S_W_ID BETWEEN 30401 AND 32300) UNION ALL SELECT * FROM
STOCK18
            WHERE (S_W_ID BETWEEN 32301 AND 34200) UNION ALL SELECT * FROM
STOCK19
            WHERE (S_W_ID BETWEEN 34201 AND 36100) UNION ALL SELECT * FROM
STOCK20
            WHERE (S_W_ID BETWEEN 36101 AND 38000) UNION ALL SELECT * FROM
STOCK21
            WHERE (S_W_ID BETWEEN 38001 AND 39900) UNION ALL SELECT * FROM
STOCK22
            WHERE (S_W_ID BETWEEN 39901 AND 41800) UNION ALL SELECT * FROM
STOCK23
            WHERE (S_W_ID BETWEEN 41801 AND 43700) UNION ALL SELECT * FROM
STOCK24
            WHERE (S_W_ID BETWEEN 43701 AND 45600) UNION ALL SELECT * FROM
STOCK25
            WHERE (S_W_ID BETWEEN 45601 AND 47500) UNION ALL SELECT * FROM
STOCK26
            WHERE (S_W_ID BETWEEN 47501 AND 49400) UNION ALL SELECT * FROM
STOCK27
            WHERE (S_W_ID BETWEEN 49401 AND 51300) UNION ALL SELECT * FROM
STOCK28
            WHERE (S_W_ID >= 51301) WITH ROW MOVEMENT;
COMMIT WORK;

connect reset;
```

GEN_CUSTOMER.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1 475   -f1 /home/tpcc/work/F1V2/tpcc_01/customer_01_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 476 950   -f1
/home/tpcc/work/F1V2/tpcc_01/customer_01_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 951 1425   -f1
/home/tpcc/work/F1V2/tpcc_01/customer_01_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1426 1900   -f1
/home/tpcc/work/F1V2/tpcc_01/customer_01_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1901 2375   -f1
/home/tpcc/work/F1V2/tpcc_02/customer_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2376 2850   -f1
/home/tpcc/work/F1V2/tpcc_02/customer_02_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2851 3325   -f1
/home/tpcc/work/F1V2/tpcc_02/customer_02_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3326 3800   -f1
/home/tpcc/work/F1V2/tpcc_02/customer_02_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3801 4275   -f1
/home/tpcc/work/F1V2/tpcc_03/customer_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4276 4750   -f1
/home/tpcc/work/F1V2/tpcc_03/customer_03_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4751 5225   -f1
/home/tpcc/work/F1V2/tpcc_03/customer_03_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5226 5700   -f1
/home/tpcc/work/F1V2/tpcc_03/customer_03_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5701 6175   -f1
/home/tpcc/work/F1V2/tpcc_04/customer_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6176 6650   -f1
/home/tpcc/work/F1V2/tpcc_04/customer_04_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6651 7125   -f1
/home/tpcc/work/F1V2/tpcc_04/customer_04_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7126 7600   -f1
/home/tpcc/work/F1V2/tpcc_04/customer_04_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7601 8075   -f1
/home/tpcc/work/F1V2/tpcc_05/customer_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8076 8550   -f1
/home/tpcc/work/F1V2/tpcc_05/customer_05_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8551 9025   -f1
/home/tpcc/work/F1V2/tpcc_05/customer_05_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9026 9500   -f1
/home/tpcc/work/F1V2/tpcc_05/customer_05_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9501 9975   -f1
/home/tpcc/work/F1V2/tpcc_06/customer_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9976 10450   -f1
/home/tpcc/work/F1V2/tpcc_06/customer_06_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10451 10925   -f1
/home/tpcc/work/F1V2/tpcc_06/customer_06_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10926 11400   -f1
/home/tpcc/work/F1V2/tpcc_06/customer_06_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 11401 11875   -f1
/home/tpcc/work/F1V2/tpcc_07/customer_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 11876 12350   -f1
/home/tpcc/work/F1V2/tpcc_07/customer_07_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12351 12825   -f1
/home/tpcc/work/F1V2/tpcc_07/customer_07_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12826 13300   -f1
/home/tpcc/work/F1V2/tpcc_07/customer_07_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 13301 13775   -f1
/home/tpcc/work/F1V2/tpcc_08/customer_08_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 13776 14250   -f1
/home/tpcc/work/F1V2/tpcc_08/customer_08_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 14251 14725   -f1
/home/tpcc/work/F1V2/tpcc_08/customer_08_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 14726 15200   -f1
/home/tpcc/work/F1V2/tpcc_08/customer_08_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 15201 15675   -f1
/home/tpcc/work/F1V2/tpcc_09/customer_09_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 15676 16150   -f1
/home/tpcc/work/F1V2/tpcc_09/customer_09_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16151 16625   -f1
/home/tpcc/work/F1V2/tpcc_09/customer_09_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16626 17100   -f1
/home/tpcc/work/F1V2/tpcc_09/customer_09_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 17101 17575   -f1
/home/tpcc/work/F1V2/tpcc_10/customer_10_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 17576 18050   -f1
/home/tpcc/work/F1V2/tpcc_10/customer_10_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 18051 18525   -f1
/home/tpcc/work/F1V2/tpcc_10/customer_10_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 18526 19000   -f1
/home/tpcc/work/F1V2/tpcc_10/customer_10_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 19001 19475   -f1
/home/tpcc/work/F1V2/tpcc_11/customer_11_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 19476 19950   -f1
/home/tpcc/work/F1V2/tpcc_11/customer_11_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 19951 20425   -f1
/home/tpcc/work/F1V2/tpcc_11/customer_11_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20426 20900   -f1
/home/tpcc/work/F1V2/tpcc_11/customer_11_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20901 21375   -f1
/home/tpcc/work/F1V2/tpcc_12/customer_12_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 21376 21850   -f1
/home/tpcc/work/F1V2/tpcc_12/customer_12_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 21851 22325   -f1
/home/tpcc/work/F1V2/tpcc_12/customer_12_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 22326 22800   -f1
/home/tpcc/work/F1V2/tpcc_12/customer_12_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 22801 23275   -f1
/home/tpcc/work/F1V2/tpcc_13/customer_13_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 23276 23750   -f1
/home/tpcc/work/F1V2/tpcc_13/customer_13_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 23751 24225   -f1
/home/tpcc/work/F1V2/tpcc_13/customer_13_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 24226 24700   -f1
/home/tpcc/work/F1V2/tpcc_13/customer_13_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 24701 25175   -f1
/home/tpcc/work/F1V2/tpcc_14/customer_14_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 25176 25650   -f1
/home/tpcc/work/F1V2/tpcc_14/customer_14_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 25651 26125   -f1
/home/tpcc/work/F1V2/tpcc_14/customer_14_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 26126 26600   -f1
/home/tpcc/work/F1V2/tpcc_14/customer_14_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 26601 27075   -f1
/home/tpcc/work/F1V2/tpcc_15/customer_15_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 27076 27550   -f1
/home/tpcc/work/F1V2/tpcc_15/customer_15_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 27551 28025   -f1
/home/tpcc/work/F1V2/tpcc_15/customer_15_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 28026 28500   -f1
/home/tpcc/work/F1V2/tpcc_15/customer_15_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 28501 28975   -f1
/home/tpcc/work/F1V2/tpcc_16/customer_16_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 28976 29450   -f1
/home/tpcc/work/F1V2/tpcc_16/customer_16_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 29451 29925   -f1
/home/tpcc/work/F1V2/tpcc_16/customer_16_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 29926 30400   -f1
/home/tpcc/work/F1V2/tpcc_16/customer_16_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 30401 30875   -f1
```

```
/home/tpcc/work/F1V2/tpcc_17/customer_17_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 30876 31350  -f1
/home/tpcc/work/F1V2/tpcc_17/customer_17_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 31351 31825  -f1
/home/tpcc/work/F1V2/tpcc_17/customer_17_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 31826 32300  -f1
/home/tpcc/work/F1V2/tpcc_17/customer_17_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 32301 32775  -f1
/home/tpcc/work/F1V2/tpcc_18/customer_18_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 32776 33250  -f1
/home/tpcc/work/F1V2/tpcc_18/customer_18_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 33251 33725  -f1
/home/tpcc/work/F1V2/tpcc_18/customer_18_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 33726 34200  -f1
/home/tpcc/work/F1V2/tpcc_18/customer_18_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 34201 34675  -f1
/home/tpcc/work/F1V2/tpcc_19/customer_19_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 34676 35150  -f1
/home/tpcc/work/F1V2/tpcc_19/customer_19_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 35151 35625  -f1
/home/tpcc/work/F1V2/tpcc_19/customer_19_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 35626 36100  -f1
/home/tpcc/work/F1V2/tpcc_19/customer_19_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 36101 36575  -f1
/home/tpcc/work/F1V2/tpcc_20/customer_20_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 36576 37050  -f1
/home/tpcc/work/F1V2/tpcc_20/customer_20_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 37051 37525  -f1
/home/tpcc/work/F1V2/tpcc_20/customer_20_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 37526 38000  -f1
/home/tpcc/work/F1V2/tpcc_20/customer_20_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 38001 38475  -f1
/home/tpcc/work/F1V2/tpcc_21/customer_21_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 38476 38950  -f1
/home/tpcc/work/F1V2/tpcc_21/customer_21_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 38951 39425  -f1
/home/tpcc/work/F1V2/tpcc_21/customer_21_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 39426 39900  -f1
/home/tpcc/work/F1V2/tpcc_21/customer_21_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 39901 40375  -f1
/home/tpcc/work/F1V2/tpcc_22/customer_22_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 40376 40850  -f1
/home/tpcc/work/F1V2/tpcc_22/customer_22_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 40851 41325  -f1
/home/tpcc/work/F1V2/tpcc_22/customer_22_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 41326 41800  -f1
/home/tpcc/work/F1V2/tpcc_22/customer_22_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 41801 42275  -f1
/home/tpcc/work/F1V2/tpcc_23/customer_23_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 42276 42750  -f1
/home/tpcc/work/F1V2/tpcc_23/customer_23_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 42751 43225  -f1
/home/tpcc/work/F1V2/tpcc_23/customer_23_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 43226 43700  -f1
/home/tpcc/work/F1V2/tpcc_23/customer_23_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 43701 44175  -f1
/home/tpcc/work/F1V2/tpcc_24/customer_24_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 44176 44650  -f1
/home/tpcc/work/F1V2/tpcc_24/customer_24_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 44651 45125  -f1
/home/tpcc/work/F1V2/tpcc_24/customer_24_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 45126 45600  -f1
/home/tpcc/work/F1V2/tpcc_24/customer_24_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 45601 46075  -f1
/home/tpcc/work/F1V2/tpcc_25/customer_25_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 46076 46550  -f1
/home/tpcc/work/F1V2/tpcc_25/customer_25_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 46551 47025  -f1
/home/tpcc/work/F1V2/tpcc_25/customer_25_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 47026 47500  -f1
/home/tpcc/work/F1V2/tpcc_25/customer_25_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 47501 47975  -f1
/home/tpcc/work/F1V2/tpcc_26/customer_26_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 47976 48450  -f1
/home/tpcc/work/F1V2/tpcc_26/customer_26_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 48451 48925  -f1
/home/tpcc/work/F1V2/tpcc_26/customer_26_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 48926 49400  -f1
/home/tpcc/work/F1V2/tpcc_26/customer_26_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 49401 49875  -f1
/home/tpcc/work/F1V2/tpcc_27/customer_27_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 49876 50350  -f1
/home/tpcc/work/F1V2/tpcc_27/customer_27_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 50351 50825  -f1
/home/tpcc/work/F1V2/tpcc_27/customer_27_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 50826 51300  -f1
/home/tpcc/work/F1V2/tpcc_27/customer_27_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 51301 51775  -f1
/home/tpcc/work/F1V2/tpcc_28/customer_28_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 51776 52250  -f1
/home/tpcc/work/F1V2/tpcc_28/customer_28_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 52251 52725  -f1
/home/tpcc/work/F1V2/tpcc_28/customer_28_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 52726 53200  -f1

/home/tpcc/work/F1V2/tpcc_28/customer_28_4.dat
```

GEN_DISTRICT.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 1 53200  -f1 /home/tpcc/work/F1V1/tpcc_00/district_1.dat
```

GEN_HISTORY.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 1 13300  -f1 /home/tpcc/work/F1V1/tpcc_00/history_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 13301 26600  -f1 /home/tpcc/work/F1V1/tpcc_00/history_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 26601 39900  -f1 /home/tpcc/work/F1V1/tpcc_00/history_3.dat

/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 39901 53200  -f1 /home/tpcc/work/F1V1/tpcc_00/history_4.dat
```

GEN_ITEM.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 5 -r 1 53200  -f1 /home/tpcc/work/F1V1/tpcc_00/item_1.dat
```

GEN_NEW_ORDER.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 1 53200  -f1 /home/tpcc/work/F1V1/tpcc_00/neworder_1.dat
```

GEN_ORDERS.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 1 475  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_1.dat -f2
/home/tpcc/work/F1V2/tpcc_01/orderline_01_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 476 950  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_2.dat -f2
/home/tpcc/work/F1V2/tpcc_01/orderline_01_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 951 1425  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_3.dat -f2
/home/tpcc/work/F1V2/tpcc_01/orderline_01_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 1426 1900  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_4.dat -f2
/home/tpcc/work/F1V2/tpcc_01/orderline_01_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 1901 2375  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_5.dat -f2
/home/tpcc/work/F1V2/tpcc_02/orderline_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 2376 2850  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_6.dat -f2
/home/tpcc/work/F1V2/tpcc_02/orderline_02_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 2851 3325  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_7.dat -f2
/home/tpcc/work/F1V2/tpcc_02/orderline_02_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 3326 3800  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_8.dat -f2
/home/tpcc/work/F1V2/tpcc_02/orderline_02_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 3801 4275  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_9.dat -f2
/home/tpcc/work/F1V2/tpcc_03/orderline_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 4276 4750  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_10.dat
-f2 /home/tpcc/work/F1V2/tpcc_03/orderline_03_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 4751 5225  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_11.dat
-f2 /home/tpcc/work/F1V2/tpcc_03/orderline_03_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 5226 5700  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_12.dat
-f2 /home/tpcc/work/F1V2/tpcc_03/orderline_03_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 5701 6175  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_13.dat
-f2 /home/tpcc/work/F1V2/tpcc_04/orderline_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 6176 6650  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_14.dat
-f2 /home/tpcc/work/F1V2/tpcc_04/orderline_04_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 6651 7125  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_15.dat
-f2 /home/tpcc/work/F1V2/tpcc_04/orderline_04_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 7126 7600  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_16.dat
-f2 /home/tpcc/work/F1V2/tpcc_04/orderline_04_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 7601 8075  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_17.dat
-f2 /home/tpcc/work/F1V2/tpcc_05/orderline_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 8076 8550  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_18.dat
-f2 /home/tpcc/work/F1V2/tpcc_05/orderline_05_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 8551 9025  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_19.dat
-f2 /home/tpcc/work/F1V2/tpcc_05/orderline_05_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 9026 9500  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_20.dat
-f2 /home/tpcc/work/F1V2/tpcc_05/orderline_05_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 9501 9975  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_21.dat
-f2 /home/tpcc/work/F1V2/tpcc_06/orderline_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 9976 10450  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_22.dat
-f2 /home/tpcc/work/F1V2/tpcc_06/orderline_06_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 10451 10925  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_23.dat
-f2 /home/tpcc/work/F1V2/tpcc_06/orderline_06_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 10926 11400  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_24.dat
-f2 /home/tpcc/work/F1V2/tpcc_06/orderline_06_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 11401 11875  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_25.dat
-f2 /home/tpcc/work/F1V2/tpcc_07/orderline_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 11876 12350  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_26.dat
-f2 /home/tpcc/work/F1V2/tpcc_07/orderline_07_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 12351 12825  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_27.dat
-f2 /home/tpcc/work/F1V2/tpcc_07/orderline_07_3.dat
```

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 12826 13300  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_28.dat
-f2 /home/tpcc/work/F1V2/tpcc_07/orderline_07_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 13301 13775  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_29.dat
-f2 /home/tpcc/work/F1V2/tpcc_08/orderline_08_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 13776 14250  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_30.dat
-f2 /home/tpcc/work/F1V2/tpcc_08/orderline_08_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 14251 14725  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_31.dat
-f2 /home/tpcc/work/F1V2/tpcc_08/orderline_08_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 14726 15200  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_32.dat
-f2 /home/tpcc/work/F1V2/tpcc_08/orderline_08_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 15201 15675  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_33.dat
-f2 /home/tpcc/work/F1V2/tpcc_09/orderline_09_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 15676 16150  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_34.dat
-f2 /home/tpcc/work/F1V2/tpcc_09/orderline_09_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 16151 16625  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_35.dat
-f2 /home/tpcc/work/F1V2/tpcc_09/orderline_09_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 16626 17100  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_36.dat
-f2 /home/tpcc/work/F1V2/tpcc_09/orderline_09_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 17101 17575  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_37.dat
-f2 /home/tpcc/work/F1V2/tpcc_10/orderline_10_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 17576 18050  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_38.dat
-f2 /home/tpcc/work/F1V2/tpcc_10/orderline_10_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 18051 18525  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_39.dat
-f2 /home/tpcc/work/F1V2/tpcc_10/orderline_10_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 18526 19000  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_40.dat
-f2 /home/tpcc/work/F1V2/tpcc_10/orderline_10_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 19001 19475  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_41.dat
-f2 /home/tpcc/work/F1V2/tpcc_11/orderline_11_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 19476 19950  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_42.dat
-f2 /home/tpcc/work/F1V2/tpcc_11/orderline_11_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 19951 20425  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_43.dat
-f2 /home/tpcc/work/F1V2/tpcc_11/orderline_11_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 20426 20900  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_44.dat
-f2 /home/tpcc/work/F1V2/tpcc_11/orderline_11_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 20901 21375  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_45.dat
-f2 /home/tpcc/work/F1V2/tpcc_12/orderline_12_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 21376 21850  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_46.dat
-f2 /home/tpcc/work/F1V2/tpcc_12/orderline_12_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 21851 22325  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_47.dat
-f2 /home/tpcc/work/F1V2/tpcc_12/orderline_12_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 22326 22800  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_48.dat
-f2 /home/tpcc/work/F1V2/tpcc_12/orderline_12_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 22801 23275  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_49.dat
-f2 /home/tpcc/work/F1V2/tpcc_13/orderline_13_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 23276 23750  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_50.dat
-f2 /home/tpcc/work/F1V2/tpcc_13/orderline_13_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 23751 24225  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_51.dat
-f2 /home/tpcc/work/F1V2/tpcc_13/orderline_13_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 24226 24700  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_52.dat
-f2 /home/tpcc/work/F1V2/tpcc_13/orderline_13_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 24701 25175  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_53.dat
-f2 /home/tpcc/work/F1V2/tpcc_14/orderline_14_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 25176 25650  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_54.dat
-f2 /home/tpcc/work/F1V2/tpcc_14/orderline_14_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 25651 26125  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_55.dat
-f2 /home/tpcc/work/F1V2/tpcc_14/orderline_14_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 26126 26600  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_56.dat
-f2 /home/tpcc/work/F1V2/tpcc_14/orderline_14_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 26601 27075  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_57.dat
-f2 /home/tpcc/work/F1V2/tpcc_15/orderline_15_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 27076 27550  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_58.dat
-f2 /home/tpcc/work/F1V2/tpcc_15/orderline_15_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 27551 28025  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_59.dat
-f2 /home/tpcc/work/F1V2/tpcc_15/orderline_15_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 28026 28500  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_60.dat
-f2 /home/tpcc/work/F1V2/tpcc_15/orderline_15_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 28501 28975  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_61.dat
-f2 /home/tpcc/work/F1V2/tpcc_16/orderline_16_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 28976 29450  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_62.dat
-f2 /home/tpcc/work/F1V2/tpcc_16/orderline_16_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 29451 29925  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_63.dat
-f2 /home/tpcc/work/F1V2/tpcc_16/orderline_16_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 29926 30400  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_64.dat
-f2 /home/tpcc/work/F1V2/tpcc_16/orderline_16_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 30401 30875  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_65.dat
-f2 /home/tpcc/work/F1V2/tpcc_17/orderline_17_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 30876 31350  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_66.dat
-f2 /home/tpcc/work/F1V2/tpcc_17/orderline_17_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 31351 31825  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_67.dat
-f2 /home/tpcc/work/F1V2/tpcc_17/orderline_17_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 31826 32300  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_68.dat
-f2 /home/tpcc/work/F1V2/tpcc_17/orderline_17_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 32301 32775  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_69.dat
-f2 /home/tpcc/work/F1V2/tpcc_18/orderline_18_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 32776 33250  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_70.dat
-f2 /home/tpcc/work/F1V2/tpcc_18/orderline_18_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 33251 33725  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_71.dat
-f2 /home/tpcc/work/F1V2/tpcc_18/orderline_18_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 33726 34200  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_72.dat
-f2 /home/tpcc/work/F1V2/tpcc_18/orderline_18_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 34201 34675  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_73.dat
-f2 /home/tpcc/work/F1V2/tpcc_19/orderline_19_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 34676 35150  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_74.dat
-f2 /home/tpcc/work/F1V2/tpcc_19/orderline_19_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 35151 35625  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_75.dat
-f2 /home/tpcc/work/F1V2/tpcc_19/orderline_19_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 35626 36100  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_76.dat
-f2 /home/tpcc/work/F1V2/tpcc_19/orderline_19_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 36101 36575  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_77.dat
-f2 /home/tpcc/work/F1V2/tpcc_20/orderline_20_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 36576 37050  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_78.dat
-f2 /home/tpcc/work/F1V2/tpcc_20/orderline_20_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 37051 37525  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_79.dat
-f2 /home/tpcc/work/F1V2/tpcc_20/orderline_20_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 37526 38000  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_80.dat
-f2 /home/tpcc/work/F1V2/tpcc_20/orderline_20_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 38001 38475  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_81.dat
-f2 /home/tpcc/work/F1V2/tpcc_21/orderline_21_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 38476 38950  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_82.dat
-f2 /home/tpcc/work/F1V2/tpcc_21/orderline_21_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 38951 39425  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_83.dat
-f2 /home/tpcc/work/F1V2/tpcc_21/orderline_21_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 39426 39900  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_84.dat
-f2 /home/tpcc/work/F1V2/tpcc_21/orderline_21_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 39901 40375  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_85.dat
-f2 /home/tpcc/work/F1V2/tpcc_22/orderline_22_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 40376 40850  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_86.dat
-f2 /home/tpcc/work/F1V2/tpcc_22/orderline_22_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 40851 41325  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_87.dat
-f2 /home/tpcc/work/F1V2/tpcc_22/orderline_22_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 41326 41800  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_88.dat
-f2 /home/tpcc/work/F1V2/tpcc_22/orderline_22_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 41801 42275  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_89.dat
-f2 /home/tpcc/work/F1V2/tpcc_23/orderline_23_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 42276 42750  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_90.dat
-f2 /home/tpcc/work/F1V2/tpcc_23/orderline_23_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 42751 43225  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_91.dat
-f2 /home/tpcc/work/F1V2/tpcc_23/orderline_23_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 43226 43700  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_92.dat
-f2 /home/tpcc/work/F1V2/tpcc_23/orderline_23_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 43701 44175  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_93.dat
-f2 /home/tpcc/work/F1V2/tpcc_24/orderline_24_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 44176 44650  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_94.dat
-f2 /home/tpcc/work/F1V2/tpcc_24/orderline_24_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 44651 45125  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_95.dat
-f2 /home/tpcc/work/F1V2/tpcc_24/orderline_24_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 45126 45600  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_96.dat
-f2 /home/tpcc/work/F1V2/tpcc_24/orderline_24_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 45601 46075  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_97.dat
-f2 /home/tpcc/work/F1V2/tpcc_25/orderline_25_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 46076 46550  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_98.dat
-f2 /home/tpcc/work/F1V2/tpcc_25/orderline_25_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 46551 47025  -f1 /home/tpcc/work/F1V1/tpcc_00/orders_99.dat
-f2 /home/tpcc/work/F1V2/tpcc_25/orderline_25_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 47026 47500  -f1
/home/tpcc/work/F1V1/tpcc_00/orders_100.dat -f2 /home/tpcc/work/F1V2/tpcc_25/orderline_25_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 47501 47975  -f1
/home/tpcc/work/F1V1/tpcc_00/orders_101.dat -f2 /home/tpcc/work/F1V2/tpcc_26/orderline_26_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 47976 48450  -f1
/home/tpcc/work/F1V1/tpcc_00/orders_102.dat -f2 /home/tpcc/work/F1V2/tpcc_26/orderline_26_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 48451 48925  -f1
/home/tpcc/work/F1V1/tpcc_00/orders_103.dat -f2 /home/tpcc/work/F1V2/tpcc_26/orderline_26_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 48926 49400  -f1
/home/tpcc/work/F1V1/tpcc_00/orders_104.dat -f2 /home/tpcc/work/F1V2/tpcc_26/orderline_26_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 49401 49875  -f1
/home/tpcc/work/F1V1/tpcc_00/orders_105.dat -f2 /home/tpcc/work/F1V2/tpcc_27/orderline_27_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 49876 50350  -f1
/home/tpcc/work/F1V1/tpcc_00/orders_106.dat -f2 /home/tpcc/work/F1V2/tpcc_27/orderline_27_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 50351 50825  -f1
/home/tpcc/work/F1V1/tpcc_00/orders_107.dat -f2 /home/tpcc/work/F1V2/tpcc_27/orderline_27_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 50826 51300  -f1
/home/tpcc/work/F1V1/tpcc_00/orders_108.dat -f2 /home/tpcc/work/F1V2/tpcc_27/orderline_27_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 51301 51775  -f1
/home/tpcc/work/F1V1/tpcc_00/orders_109.dat -f2 /home/tpcc/work/F1V2/tpcc_28/orderline_28_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 51776 52250  -f1
/home/tpcc/work/F1V1/tpcc_00/orders_110.dat -f2 /home/tpcc/work/F1V2/tpcc_28/orderline_28_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 52251 52725  -f1
/home/tpcc/work/F1V1/tpcc_00/orders_111.dat -f2 /home/tpcc/work/F1V2/tpcc_28/orderline_28_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 52726 53200  -f1

/home/tpcc/work/F1V1/tpcc_00/orders_112.dat -f2 /home/tpcc/work/F1V2/tpcc_28/orderline_28_4.dat
```

GEN_STOCK.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1 475  -f1 /home/tpcc/work/F1V1/tpcc_01/stock_01_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 476 950  -f1 /home/tpcc/work/F1V1/tpcc_01/stock_01_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 951 1425  -f1 /home/tpcc/work/F1V1/tpcc_01/stock_01_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1426 1900  -f1 /home/tpcc/work/F1V1/tpcc_01/stock_01_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1901 2375  -f1 /home/tpcc/work/F1V1/tpcc_02/stock_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 2376 2850  -f1 /home/tpcc/work/F1V1/tpcc_02/stock_02_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 2851 3325  -f1 /home/tpcc/work/F1V1/tpcc_02/stock_02_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 3326 3800  -f1 /home/tpcc/work/F1V1/tpcc_02/stock_02_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 3801 4275  -f1 /home/tpcc/work/F1V1/tpcc_03/stock_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 4276 4750  -f1 /home/tpcc/work/F1V1/tpcc_03/stock_03_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 4751 5225  -f1 /home/tpcc/work/F1V1/tpcc_03/stock_03_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 5226 5700  -f1 /home/tpcc/work/F1V1/tpcc_03/stock_03_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 5701 6175  -f1 /home/tpcc/work/F1V1/tpcc_04/stock_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 6176 6650  -f1 /home/tpcc/work/F1V1/tpcc_04/stock_04_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 6651 7125  -f1 /home/tpcc/work/F1V1/tpcc_04/stock_04_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 7126 7600  -f1 /home/tpcc/work/F1V1/tpcc_04/stock_04_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 7601 8075  -f1 /home/tpcc/work/F1V1/tpcc_05/stock_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 8076 8550  -f1 /home/tpcc/work/F1V1/tpcc_05/stock_05_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 8551 9025  -f1 /home/tpcc/work/F1V1/tpcc_05/stock_05_3.dat
```

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 9026 9500   -f1 /home/tpcc/work/F1V1/tpcc_05/stock_05_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 9501 9975   -f1 /home/tpcc/work/F1V1/tpcc_06/stock_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 9976 10450   -f1
/home/tpcc/work/F1V1/tpcc_06/stock_06_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 10451 10925   -f1
/home/tpcc/work/F1V1/tpcc_06/stock_06_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 10926 11400   -f1
/home/tpcc/work/F1V1/tpcc_06/stock_06_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 11401 11875   -f1
/home/tpcc/work/F1V1/tpcc_07/stock_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 11876 12350   -f1
/home/tpcc/work/F1V1/tpcc_07/stock_07_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 12351 12825   -f1
/home/tpcc/work/F1V1/tpcc_07/stock_07_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 12826 13300   -f1
/home/tpcc/work/F1V1/tpcc_07/stock_07_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 13301 13775   -f1
/home/tpcc/work/F1V1/tpcc_08/stock_08_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 13776 14250   -f1
/home/tpcc/work/F1V1/tpcc_08/stock_08_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 14251 14725   -f1
/home/tpcc/work/F1V1/tpcc_08/stock_08_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 14726 15200   -f1
/home/tpcc/work/F1V1/tpcc_08/stock_08_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 15201 15675   -f1
/home/tpcc/work/F1V1/tpcc_09/stock_09_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 15676 16150   -f1
/home/tpcc/work/F1V1/tpcc_09/stock_09_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 16151 16625   -f1
/home/tpcc/work/F1V1/tpcc_09/stock_09_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 16626 17100   -f1
/home/tpcc/work/F1V1/tpcc_09/stock_09_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 17101 17575   -f1
/home/tpcc/work/F1V1/tpcc_10/stock_10_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 17576 18050   -f1
/home/tpcc/work/F1V1/tpcc_10/stock_10_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 18051 18525   -f1
/home/tpcc/work/F1V1/tpcc_10/stock_10_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 18526 19000   -f1
/home/tpcc/work/F1V1/tpcc_10/stock_10_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 19001 19475   -f1
/home/tpcc/work/F1V1/tpcc_11/stock_11_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 19476 19950   -f1
/home/tpcc/work/F1V1/tpcc_11/stock_11_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 19951 20425   -f1
/home/tpcc/work/F1V1/tpcc_11/stock_11_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 20426 20900   -f1
/home/tpcc/work/F1V1/tpcc_11/stock_11_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 20901 21375   -f1
/home/tpcc/work/F1V1/tpcc_12/stock_12_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 21376 21850   -f1
/home/tpcc/work/F1V1/tpcc_12/stock_12_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 21851 22325   -f1
/home/tpcc/work/F1V1/tpcc_12/stock_12_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 22326 22800   -f1
/home/tpcc/work/F1V1/tpcc_12/stock_12_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 22801 23275   -f1
/home/tpcc/work/F1V1/tpcc_13/stock_13_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 23276 23750   -f1
/home/tpcc/work/F1V1/tpcc_13/stock_13_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 23751 24225   -f1
/home/tpcc/work/F1V1/tpcc_13/stock_13_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 24226 24700   -f1
/home/tpcc/work/F1V1/tpcc_13/stock_13_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 24701 25175   -f1
/home/tpcc/work/F1V1/tpcc_14/stock_14_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 25176 25650   -f1
/home/tpcc/work/F1V1/tpcc_14/stock_14_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 25651 26125   -f1
/home/tpcc/work/F1V1/tpcc_14/stock_14_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 26126 26600   -f1
/home/tpcc/work/F1V1/tpcc_14/stock_14_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 26601 27075   -f1
/home/tpcc/work/F1V1/tpcc_15/stock_15_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 27076 27550   -f1
/home/tpcc/work/F1V1/tpcc_15/stock_15_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 27551 28025   -f1
/home/tpcc/work/F1V1/tpcc_15/stock_15_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 28026 28500   -f1
/home/tpcc/work/F1V1/tpcc_15/stock_15_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 28501 28975   -f1
/home/tpcc/work/F1V1/tpcc_16/stock_16_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 28976 29450   -f1
/home/tpcc/work/F1V1/tpcc_16/stock_16_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 29451 29925   -f1
/home/tpcc/work/F1V1/tpcc_16/stock_16_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 29926 30400   -f1
/home/tpcc/work/F1V1/tpcc_16/stock_16_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 30401 30875   -f1
/home/tpcc/work/F1V1/tpcc_17/stock_17_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 30876 31350   -f1
/home/tpcc/work/F1V1/tpcc_17/stock_17_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 31351 31825   -f1
/home/tpcc/work/F1V1/tpcc_17/stock_17_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 31826 32300   -f1
/home/tpcc/work/F1V1/tpcc_17/stock_17_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 32301 32775   -f1
/home/tpcc/work/F1V1/tpcc_18/stock_18_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 32776 33250   -f1
/home/tpcc/work/F1V1/tpcc_18/stock_18_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 33251 33725   -f1
/home/tpcc/work/F1V1/tpcc_18/stock_18_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 33726 34200   -f1
/home/tpcc/work/F1V1/tpcc_18/stock_18_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 34201 34675   -f1
/home/tpcc/work/F1V1/tpcc_19/stock_19_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 34676 35150   -f1
/home/tpcc/work/F1V1/tpcc_19/stock_19_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 35151 35625   -f1
/home/tpcc/work/F1V1/tpcc_19/stock_19_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 35626 36100   -f1
/home/tpcc/work/F1V1/tpcc_19/stock_19_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 36101 36575   -f1
/home/tpcc/work/F1V1/tpcc_20/stock_20_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 36576 37050   -f1
/home/tpcc/work/F1V1/tpcc_20/stock_20_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 37051 37525   -f1
/home/tpcc/work/F1V1/tpcc_20/stock_20_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 37526 38000   -f1
/home/tpcc/work/F1V1/tpcc_20/stock_20_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 38001 38475   -f1
/home/tpcc/work/F1V1/tpcc_21/stock_21_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 38476 38950   -f1
/home/tpcc/work/F1V1/tpcc_21/stock_21_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 38951 39425   -f1
/home/tpcc/work/F1V1/tpcc_21/stock_21_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 39426 39900   -f1
/home/tpcc/work/F1V1/tpcc_21/stock_21_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 39901 40375   -f1
/home/tpcc/work/F1V1/tpcc_22/stock_22_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 40376 40850   -f1
/home/tpcc/work/F1V1/tpcc_22/stock_22_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 40851 41325   -f1
/home/tpcc/work/F1V1/tpcc_22/stock_22_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 41326 41800   -f1
/home/tpcc/work/F1V1/tpcc_22/stock_22_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 41801 42275   -f1
/home/tpcc/work/F1V1/tpcc_23/stock_23_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 42276 42750   -f1
/home/tpcc/work/F1V1/tpcc_23/stock_23_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 42751 43225   -f1
/home/tpcc/work/F1V1/tpcc_23/stock_23_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 43226 43700   -f1
/home/tpcc/work/F1V1/tpcc_23/stock_23_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 43701 44175   -f1
/home/tpcc/work/F1V1/tpcc_24/stock_24_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 44176 44650   -f1
/home/tpcc/work/F1V1/tpcc_24/stock_24_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 44651 45125   -f1
/home/tpcc/work/F1V1/tpcc_24/stock_24_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 45126 45600   -f1
/home/tpcc/work/F1V1/tpcc_24/stock_24_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 45601 46075   -f1
/home/tpcc/work/F1V1/tpcc_25/stock_25_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 46076 46550   -f1
/home/tpcc/work/F1V1/tpcc_25/stock_25_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 46551 47025   -f1
/home/tpcc/work/F1V1/tpcc_25/stock_25_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 47026 47500   -f1
/home/tpcc/work/F1V1/tpcc_25/stock_25_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 47501 47975   -f1
/home/tpcc/work/F1V1/tpcc_26/stock_26_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 47976 48450   -f1
/home/tpcc/work/F1V1/tpcc_26/stock_26_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 48451 48925   -f1
/home/tpcc/work/F1V1/tpcc_26/stock_26_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 48926 49400   -f1
/home/tpcc/work/F1V1/tpcc_26/stock_26_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 49401 49875   -f1
/home/tpcc/work/F1V1/tpcc_27/stock_27_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 49876 50350   -f1
/home/tpcc/work/F1V1/tpcc_27/stock_27_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 50351 50825   -f1
/home/tpcc/work/F1V1/tpcc_27/stock_27_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 50826 51300   -f1
/home/tpcc/work/F1V1/tpcc_27/stock_27_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 51301 51775   -f1
/home/tpcc/work/F1V1/tpcc_28/stock_28_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 51776 52250   -f1
/home/tpcc/work/F1V1/tpcc_28/stock_28_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 52251 52725   -f1
/home/tpcc/work/F1V1/tpcc_28/stock_28_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 52726 53200   -f1

/home/tpcc/work/F1V1/tpcc_28/stock_28_4.dat
```

GEN_WAREHOUSE.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 1 53200   -f1 /home/tpcc/work/F1V1/tpcc_00/warehouse_1.dat
```

LOAD_CUSTOMER_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_01/customer_01_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_01/customer_01_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_01/customer_01_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_01/customer_01_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_02/customer_02_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_02/customer_02_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_02/customer_02_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_02/customer_02_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_03/customer_03_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_03/customer_03_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_03/customer_03_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_03/customer_03_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_04/customer_04_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_04/customer_04_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_04/customer_04_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_04/customer_04_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_05/customer_05_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_05/customer_05_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_05/customer_05_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_05/customer_05_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_06/customer_06_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_06/customer_06_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_06/customer_06_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_06/customer_06_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_07/customer_07_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_07/customer_07_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_07/customer_07_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_07/customer_07_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_08/customer_08_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_08/customer_08_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_08/customer_08_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_08/customer_08_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER9 ACTIVATE NOT LOGGED INITIALLY;
```

```
IMPORT FROM /home/tpcc/work/F1V2/tpcc_09/customer_09_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_09/customer_09_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_09/customer_09_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_09/customer_09_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_10/customer_10_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_10/customer_10_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_10/customer_10_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_10/customer_10_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_11/customer_11_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_11/customer_11_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_11/customer_11_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_11/customer_11_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_12/customer_12_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_12/customer_12_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_12/customer_12_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_12/customer_12_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_13/customer_13_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_13/customer_13_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_13/customer_13_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_13/customer_13_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_14/customer_14_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_14/customer_14_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_14/customer_14_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_14/customer_14_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_15/customer_15_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_15/customer_15_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_15/customer_15_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_15/customer_15_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_16/customer_16_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_16/customer_16_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_16/customer_16_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_16/customer_16_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_17/customer_17_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER17 ACTIVATE NOT LOGGED INITIALLY;
```

```
IMPORT FROM /home/tpcc/work/F1V2/tpcc_17/customer_17_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_17/customer_17_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_17/customer_17_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_18/customer_18_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_18/customer_18_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_18/customer_18_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_18/customer_18_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_19/customer_19_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_19/customer_19_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_19/customer_19_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_19/customer_19_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_20/customer_20_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_20/customer_20_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_20/customer_20_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_20/customer_20_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_21/customer_21_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_21/customer_21_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_21/customer_21_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_21/customer_21_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_22/customer_22_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_22/customer_22_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_22/customer_22_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_22/customer_22_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_23/customer_23_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_23/customer_23_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_23/customer_23_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_23/customer_23_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_24/customer_24_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_24/customer_24_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_24/customer_24_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_24/customer_24_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_25/customer_25_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_25/customer_25_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER25 ACTIVATE NOT LOGGED INITIALLY;
```

```
IMPORT FROM /home/tpcc/work/F1V2/tpcc_25/customer_25_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_25/customer_25_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_26/customer_26_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_26/customer_26_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_26/customer_26_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_26/customer_26_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_27/customer_27_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_27/customer_27_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_27/customer_27_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_27/customer_27_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_28/customer_28_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_28/customer_28_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_28/customer_28_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE CUSTOMER28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_28/customer_28_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO CUSTOMER28;
COMMIT WORK;

CONNECT RESET;
```

LOADE_DISTRICT_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE DISTRICT ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/district_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 532000 INSERT INTO DISTRICT;
COMMIT WORK;

CONNECT RESET;
```

LOAD_HISTORY_ALL.ddl

```
connect to TPCC in share mode;
LOAD FROM /home/tpcc/work/F1V1/tpcc_00/history_1.dat, /home/tpcc/work/F1V1/tpcc_00/history_2.dat,
/home/tpcc/work/F1V1/tpcc_00/history_3.dat, /home/tpcc/work/F1V1/tpcc_00/history_4.dat OF DEL
MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE REPLACE INTO HISTORY
NONRECOVERABLE DATA BUFFER 40000 CPU_PARALLELISM 8 ;

connect reset;
```

LOAD_ITEM_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ITEM ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/item_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 100000 INSERT INTO ITEM;
COMMIT WORK;

CONNECT RESET;
```

LOAD_NEW-ORDER_ALL.ddl

```
connect to TPCC in share mode;
LOAD FROM /home/tpcc/work/F1V1/tpcc_00/neworder_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE REPLACE INTO NEW_ORDER NONRECOVERABLE DATA BUFFER
40000 CPU_PARALLELISM 8 ;

connect reset;
```

LOAD_ORDERS_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_5.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_6.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_7.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_8.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_9.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_10.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_11.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_12.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_13.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_14.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_15.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_16.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_17.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_18.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_19.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_20.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_21.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_22.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_23.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_24.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_25.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_26.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_27.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_28.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_29.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_30.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_31.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_32.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_33.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_34.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_35.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_36.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_37.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_38.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_39.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_40.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_41.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_42.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_43.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_44.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_45.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_46.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_47.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_48.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_49.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_50.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_51.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_52.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_53.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_54.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_55.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_56.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_57.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_58.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_59.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_60.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_61.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_62.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_63.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_64.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_65.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_66.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_67.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_68.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_69.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_70.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_71.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_72.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_73.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_74.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_75.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_76.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_77.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_78.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_79.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_80.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_81.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_82.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_83.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_84.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_85.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_86.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_87.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_88.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_89.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_90.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_91.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_92.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_93.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_94.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_95.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_96.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_97.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_98.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_99.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_100.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_101.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_102.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_103.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_104.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_105.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_106.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_107.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_108.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_109.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_110.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_111.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDERS ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/orders_112.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 14250000 INSERT INTO ORDERS;
COMMIT WORK;

CONNECT RESET;
```

LOAD_ORDER_LINE_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_01/orderline_01_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_01/orderline_01_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_01/orderline_01_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_01/orderline_01_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_02/orderline_02_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_02/orderline_02_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_02/orderline_02_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_02/orderline_02_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_03/orderline_03_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_03/orderline_03_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_03/orderline_03_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_03/orderline_03_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_04/orderline_04_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_04/orderline_04_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_04/orderline_04_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_04/orderline_04_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_05/orderline_05_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_05/orderline_05_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_05/orderline_05_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_05/orderline_05_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_06/orderline_06_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_06/orderline_06_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_06/orderline_06_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_06/orderline_06_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_07/orderline_07_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_07/orderline_07_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE7;
```

```
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_07/orderline_07_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_07/orderline_07_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_08/orderline_08_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_08/orderline_08_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_08/orderline_08_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_08/orderline_08_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_09/orderline_09_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_09/orderline_09_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_09/orderline_09_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_09/orderline_09_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_10/orderline_10_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_10/orderline_10_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_10/orderline_10_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_10/orderline_10_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_11/orderline_11_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_11/orderline_11_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_11/orderline_11_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_11/orderline_11_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_12/orderline_12_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_12/orderline_12_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_12/orderline_12_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_12/orderline_12_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_13/orderline_13_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_13/orderline_13_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_13/orderline_13_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_13/orderline_13_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_14/orderline_14_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_14/orderline_14_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_14/orderline_14_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_14/orderline_14_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_15/orderline_15_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_15/orderline_15_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_15/orderline_15_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE15;
```

```
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_15/orderline_15_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_16/orderline_16_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_16/orderline_16_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_16/orderline_16_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_16/orderline_16_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_17/orderline_17_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_17/orderline_17_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_17/orderline_17_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_17/orderline_17_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_18/orderline_18_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_18/orderline_18_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_18/orderline_18_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_18/orderline_18_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_19/orderline_19_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_19/orderline_19_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_19/orderline_19_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_19/orderline_19_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_20/orderline_20_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_20/orderline_20_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_20/orderline_20_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_20/orderline_20_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_21/orderline_21_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_21/orderline_21_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_21/orderline_21_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_21/orderline_21_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_22/orderline_22_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_22/orderline_22_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_22/orderline_22_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_22/orderline_22_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_23/orderline_23_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_23/orderline_23_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_23/orderline_23_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_23/orderline_23_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE23;
```

```
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_24/orderline_24_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_24/orderline_24_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_24/orderline_24_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_24/orderline_24_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_25/orderline_25_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_25/orderline_25_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_25/orderline_25_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_25/orderline_25_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_26/orderline_26_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_26/orderline_26_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_26/orderline_26_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_26/orderline_26_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_27/orderline_27_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_27/orderline_27_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_27/orderline_27_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_27/orderline_27_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_28/orderline_28_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_28/orderline_28_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_28/orderline_28_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE ORDER_LINE28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V2/tpcc_28/orderline_28_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 171000000 INSERT INTO ORDER_LINE28;
COMMIT WORK;

CONNECT RESET;
```

LOAD_STOCK_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_01/stock_01_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_01/stock_01_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_01/stock_01_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_01/stock_01_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_02/stock_02_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_02/stock_02_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_02/stock_02_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_02/stock_02_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_03/stock_03_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_03/stock_03_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_03/stock_03_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_03/stock_03_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_04/stock_04_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_04/stock_04_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_04/stock_04_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_04/stock_04_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_05/stock_05_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_05/stock_05_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_05/stock_05_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_05/stock_05_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_06/stock_06_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_06/stock_06_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_06/stock_06_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_06/stock_06_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_07/stock_07_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_07/stock_07_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_07/stock_07_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_07/stock_07_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_08/stock_08_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_08/stock_08_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_08/stock_08_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_08/stock_08_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_09/stock_09_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_09/stock_09_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_09/stock_09_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_09/stock_09_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_10/stock_10_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_10/stock_10_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_10/stock_10_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_10/stock_10_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_11/stock_11_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_11/stock_11_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_11/stock_11_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_11/stock_11_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
ALTER TABLE STOCK12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_12/stock_12_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_12/stock_12_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_12/stock_12_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_12/stock_12_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_13/stock_13_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_13/stock_13_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_13/stock_13_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_13/stock_13_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_14/stock_14_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_14/stock_14_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_14/stock_14_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_14/stock_14_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_15/stock_15_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_15/stock_15_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_15/stock_15_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_15/stock_15_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_16/stock_16_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_16/stock_16_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_16/stock_16_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_16/stock_16_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_17/stock_17_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_17/stock_17_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_17/stock_17_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_17/stock_17_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_18/stock_18_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_18/stock_18_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_18/stock_18_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_18/stock_18_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_19/stock_19_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_19/stock_19_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_19/stock_19_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_19/stock_19_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_20/stock_20_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
ALTER TABLE STOCK20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_20/stock_20_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_20/stock_20_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_20/stock_20_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_21/stock_21_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_21/stock_21_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_21/stock_21_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_21/stock_21_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_22/stock_22_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_22/stock_22_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_22/stock_22_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_22/stock_22_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_23/stock_23_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_23/stock_23_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_23/stock_23_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_23/stock_23_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_24/stock_24_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_24/stock_24_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_24/stock_24_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_24/stock_24_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_25/stock_25_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_25/stock_25_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_25/stock_25_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_25/stock_25_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_26/stock_26_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_26/stock_26_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_26/stock_26_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_26/stock_26_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_27/stock_27_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_27/stock_27_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_27/stock_27_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_27/stock_27_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_28/stock_28_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_28/stock_28_2.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
```

```
ALTER TABLE STOCK28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_28/stock_28_3.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE STOCK28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_28/stock_28_4.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 47500000 INSERT INTO STOCK28;
COMMIT WORK;

CONNECT RESET;
```

LOAD_WAREHOUSE_ALL.ddl

```
CONNECT TO TPCC IN SHARE MODE;
ALTER TABLE WAREHOUSE ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /home/tpcc/work/F1V1/tpcc_00/warehouse_1.dat OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50 COMMITCOUNT 53200 INSERT INTO WAREHOUSE;
COMMIT WORK;

CONNECT RESET;
```

RNST_CUSTOMER.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER28 AND INDEXES ALL;
COMMIT WORK;

connect reset;
```

RNST_DISTRICT.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.DISTRICT AND INDEXES ALL;
COMMIT WORK;

connect reset;
```

RNST_HISTORY.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.HISTORY AND INDEXES ALL;
COMMIT WORK;

connect reset;
```

RNST_ITEM.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ITEM AND INDEXES ALL;
COMMIT WORK;

connect reset;
```

RNST_NEW_ORDER.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.NEW_ORDER AND INDEXES ALL;
COMMIT WORK;

connect reset;
```

RNST_ORDERS.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDERS AND INDEXES ALL;
COMMIT WORK;

connect reset;
```

RNST_ORDER_LINE.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE28 AND INDEXES ALL;
COMMIT WORK;

connect reset;
```

RNST_STOCK.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK6 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK7 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK8 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK9 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK10 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK11 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK12 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK13 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK14 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK15 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK16 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK17 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

```
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK18 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK19 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK20 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK21 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK22 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK23 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK24 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK25 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK26 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK27 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.STOCK28 AND INDEXES ALL;
COMMIT WORK;

connect reset;
```

RNST_WAREHOUSE.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE TPCC.WAREHOUSE AND INDEXES ALL;
COMMIT WORK;

connect reset;
```

SORT_ORDERS.sh

```
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_1.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_1.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_1.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_1.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_1.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_2.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_2.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_2.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_2.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_2.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_2.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_3.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_3.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_3.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_3.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_3.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_3.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_4.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_4.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_4.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_4.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_4.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_4.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_5.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_5.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_5.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_5.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_5.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_5.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_6.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_6.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_6.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_6.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_6.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_6.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_7.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_7.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_7.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_7.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_7.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_7.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_8.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_8.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_8.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_8.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_8.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_8.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_9.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_9.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_9.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_9.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_9.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_9.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_10.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_10.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_10.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_10.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_10.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_10.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_11.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_11.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_11.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_11.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_11.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_11.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_12.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_12.dat.sorted
if [ $? = 0 ]
then
```

```
 mv /home/tpcc/work/F1V1/tpcc_00/orders_12.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_12.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_12.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_12.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_13.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_13.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_13.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_13.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_13.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_13.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_14.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_14.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_14.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_14.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_14.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_14.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_15.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_15.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_15.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_15.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_15.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_15.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_16.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_16.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_16.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_16.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_16.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_16.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_17.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_17.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_17.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_17.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_17.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_17.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_18.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_18.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_18.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_18.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_18.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_18.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_19.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_19.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_19.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_19.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_19.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_19.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_20.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_20.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_20.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_20.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_20.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_20.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_21.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_21.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_21.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_21.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_21.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_21.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_22.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_22.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_22.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_22.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_22.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_22.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_23.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_23.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_23.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_23.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_23.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_23.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_24.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_24.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_24.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_24.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_24.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_24.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_25.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_25.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_25.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_25.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_25.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_25.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_26.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_26.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_26.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_26.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_26.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_26.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_27.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_27.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_27.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_27.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_27.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_27.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_28.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_28.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_28.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_28.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_28.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_28.dat)"
 echo "exists and that sufficient free space is available"
```

```
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_29.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_29.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_29.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_29.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_29.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_29.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_30.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_30.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_30.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_30.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_30.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_30.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_31.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_31.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_31.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_31.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_31.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_31.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_32.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_32.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_32.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_32.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_32.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_32.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_33.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_33.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_33.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_33.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_33.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_33.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_34.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_34.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_34.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_34.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_34.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_34.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_35.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_35.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_35.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_35.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_35.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_35.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_36.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_36.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_36.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_36.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_36.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_36.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_37.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_37.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_37.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_37.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_37.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_37.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_38.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_38.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_38.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_38.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_38.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_38.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_39.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_39.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_39.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_39.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_39.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_39.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_40.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_40.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_40.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_40.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_40.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_40.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_41.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_41.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_41.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_41.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_41.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_41.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_42.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_42.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_42.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_42.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_42.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_42.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_43.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_43.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_43.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_43.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_43.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_43.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_44.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_44.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_44.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_44.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_44.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_44.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_45.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_45.dat.sorted
if [ $? = 0 ]
then
```

```
 mv /home/tpcc/work/F1V1/tpcc_00/orders_45.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_45.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_45.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_45.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_46.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_46.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_46.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_46.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_46.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_46.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_47.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_47.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_47.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_47.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_47.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_47.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_48.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_48.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_48.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_48.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_48.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_48.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_49.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_49.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_49.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_49.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_49.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_49.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_50.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_50.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_50.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_50.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_50.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_50.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_51.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_51.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_51.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_51.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_51.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_51.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_52.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_52.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_52.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_52.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_52.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_52.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_53.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_53.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_53.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_53.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_53.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_53.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_54.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_54.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_54.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_54.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_54.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_54.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_55.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_55.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_55.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_55.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_55.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_55.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_56.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_56.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_56.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_56.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_56.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_56.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_57.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_57.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_57.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_57.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_57.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_57.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_58.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_58.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_58.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_58.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_58.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_58.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_59.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_59.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_59.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_59.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_59.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_59.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_60.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_60.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_60.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_60.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_60.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_60.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_61.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_61.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_61.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_61.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_61.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_61.dat)"
 echo "exists and that sufficient free space is available"
```

```
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_62.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_62.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_62.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_62.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_62.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_62.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_63.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_63.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_63.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_63.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_63.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_63.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_64.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_64.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_64.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_64.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_64.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_64.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_65.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_65.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_65.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_65.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_65.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_65.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_66.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_66.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_66.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_66.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_66.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_66.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_67.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_67.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_67.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_67.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_67.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_67.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_68.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_68.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_68.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_68.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_68.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_68.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_69.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_69.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_69.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_69.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_69.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_69.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_70.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_70.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_70.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_70.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_70.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_70.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_71.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_71.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_71.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_71.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_71.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_71.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_72.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_72.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_72.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_72.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_72.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_72.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_73.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_73.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_73.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_73.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_73.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_73.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_74.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_74.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_74.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_74.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_74.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_74.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_75.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_75.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_75.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_75.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_75.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_75.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_76.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_76.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_76.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_76.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_76.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_76.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_77.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_77.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_77.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_77.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_77.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_77.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_78.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_78.dat.sorted
if [ $? = 0 ]
then
```

```
 mv /home/tpcc/work/F1V1/tpcc_00/orders_78.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_78.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_78.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_78.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_79.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_79.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_79.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_79.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_79.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_79.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_80.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_80.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_80.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_80.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_80.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_80.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_81.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_81.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_81.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_81.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_81.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_81.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_82.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_82.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_82.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_82.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_82.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_82.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_83.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_83.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_83.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_83.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_83.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_83.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_84.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_84.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_84.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_84.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_84.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_84.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_85.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_85.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_85.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_85.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_85.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_85.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_86.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_86.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_86.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_86.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_86.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_86.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_87.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_87.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_87.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_87.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_87.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_87.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_88.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_88.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_88.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_88.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_88.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_88.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_89.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_89.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_89.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_89.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_89.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_89.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_90.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_90.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_90.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_90.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_90.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_90.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_91.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_91.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_91.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_91.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_91.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_91.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_92.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_92.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_92.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_92.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_92.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_92.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_93.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_93.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_93.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_93.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_93.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_93.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_94.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_94.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_94.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_94.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_94.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_94.dat)"
 echo "exists and that sufficient free space is available"
```

```
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_95.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_95.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_95.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_95.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_95.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_95.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_96.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_96.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_96.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_96.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_96.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_96.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_97.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_97.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_97.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_97.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_97.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_97.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_98.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_98.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_98.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_98.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_98.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_98.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_99.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_99.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_99.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_99.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_99.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_99.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_100.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_100.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_100.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_100.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_100.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_100.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_101.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_101.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_101.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_101.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_101.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_101.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_102.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_102.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_102.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_102.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_102.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_102.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_103.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_103.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_103.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_103.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_103.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_103.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_104.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_104.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_104.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_104.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_104.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_104.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_105.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_105.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_105.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_105.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_105.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_105.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_106.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_106.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_106.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_106.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_106.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_106.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_107.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_107.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_107.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_107.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_107.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_107.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_108.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_108.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_108.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_108.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_108.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_108.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_109.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_109.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_109.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_109.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_109.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_109.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_110.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_110.dat.sorted
if [ $? = 0 ]
then
  mv /home/tpcc/work/F1V1/tpcc_00/orders_110.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_110.dat
else
  echo "**** WARNING ****"
  echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_110.dat FAILED."
  echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_110.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_111.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_111.dat.sorted
if [ $? = 0 ]
then
```

```
 mv /home/tpcc/work/F1V1/tpcc_00/orders_111.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_111.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_111.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_111.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."
fi
sort -t'|' -T /home/tpcc/work/F1V1/tpcc_00 +5n +6n +7n /home/tpcc/work/F1V1/tpcc_00/orders_112.dat >
/home/tpcc/work/F1V1/tpcc_00/orders_112.dat.sorted
if [ $? = 0 ]
then
 mv /home/tpcc/work/F1V1/tpcc_00/orders_112.dat.sorted /home/tpcc/work/F1V1/tpcc_00/orders_112.dat
else
 echo "**** WARNING ****"
 echo "Sort of /home/tpcc/work/F1V1/tpcc_00/orders_112.dat FAILED."
 echo "Please ensure that the source file (/home/tpcc/work/F1V1/tpcc_00/orders_112.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/home/tpcc/work/F1V1/tpcc_00)."

fi
```

alter_tablespace.ddl

```
-------------------------------------------------------------------------------
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
-------------------------------------------------------------------------------

-- Set Bufferpools For Tablespaces
connect to tpcc;

alter tablespace TS_WARE_01 bufferpool WAR;
alter tablespace TS_DIST_01 bufferpool DIS;
alter tablespace TS_ITEM_01 bufferpool ITM;
alter tablespace TS_HISTORY_01 bufferpool HST;
alter tablespace TS_NEWORD_01 bufferpool NEW;
alter tablespcae IS_NEWORD_01 bufferpool NEW_I;
alter tablespace TS_ORDER_01 bufferpool ORD;
alter tablespace IS_ORDER_01 bufferpool ORD_I;

alter tablespace TS_ORDERLINE_01 bufferpool OLN;
alter tablespace TS_ORDERLINE_02 bufferpool OLN;
alter tablespace TS_ORDERLINE_03 bufferpool OLN;
alter tablespace TS_ORDERLINE_04 bufferpool OLN;
alter tablespace TS_ORDERLINE_05 bufferpool OLN;
alter tablespace TS_ORDERLINE_06 bufferpool OLN;
alter tablespace TS_ORDERLINE_07 bufferpool OLN;
alter tablespace TS_ORDERLINE_08 bufferpool OLN;
alter tablespace TS_ORDERLINE_09 bufferpool OLN;
alter tablespace TS_ORDERLINE_10 bufferpool OLN;
alter tablespace TS_ORDERLINE_11 bufferpool OLN;
alter tablespace TS_ORDERLINE_12 bufferpool OLN;
alter tablespace TS_ORDERLINE_13 bufferpool OLN;
alter tablespace TS_ORDERLINE_14 bufferpool OLN;
alter tablespace TS_ORDERLINE_15 bufferpool OLN;
alter tablespace TS_ORDERLINE_16 bufferpool OLN;
alter tablespace TS_ORDERLINE_17 bufferpool OLN;
alter tablespace TS_ORDERLINE_18 bufferpool OLN;
alter tablespace TS_ORDERLINE_19 bufferpool OLN;
alter tablespace TS_ORDERLINE_20 bufferpool OLN;
alter tablespace TS_ORDERLINE_21 bufferpool OLN;
alter tablespace TS_ORDERLINE_22 bufferpool OLN;
alter tablespace TS_ORDERLINE_23 bufferpool OLN;
alter tablespace TS_ORDERLINE_24 bufferpool OLN;
alter tablespace TS_ORDERLINE_25 bufferpool OLN;
alter tablespace TS_ORDERLINE_26 bufferpool OLN;
alter tablespace TS_ORDERLINE_27 bufferpool OLN;
alter tablespace TS_ORDERLINE_28 bufferpool OLN;

alter tablespace TS_CUSTOMER_01 bufferpool CST;
alter tablespace TS_CUSTOMER_02 bufferpool CST;
alter tablespace TS_CUSTOMER_03 bufferpool CST;
alter tablespace TS_CUSTOMER_04 bufferpool CST;
alter tablespace TS_CUSTOMER_05 bufferpool CST;
alter tablespace TS_CUSTOMER_06 bufferpool CST;
alter tablespace TS_CUSTOMER_07 bufferpool CST;
alter tablespace TS_CUSTOMER_08 bufferpool CST;
alter tablespace TS_CUSTOMER_09 bufferpool CST;
alter tablespace TS_CUSTOMER_10 bufferpool CST;
alter tablespace TS_CUSTOMER_11 bufferpool CST;
alter tablespace TS_CUSTOMER_12 bufferpool CST;
alter tablespace TS_CUSTOMER_13 bufferpool CST;
alter tablespace TS_CUSTOMER_14 bufferpool CST;
alter tablespace TS_CUSTOMER_15 bufferpool CST;
alter tablespace TS_CUSTOMER_16 bufferpool CST;
alter tablespace TS_CUSTOMER_17 bufferpool CST;
alter tablespace TS_CUSTOMER_18 bufferpool CST;
alter tablespace TS_CUSTOMER_19 bufferpool CST;
alter tablespace TS_CUSTOMER_20 bufferpool CST;
alter tablespace TS_CUSTOMER_21 bufferpool CST;
alter tablespace TS_CUSTOMER_22 bufferpool CST;
alter tablespace TS_CUSTOMER_23 bufferpool CST;
alter tablespace TS_CUSTOMER_24 bufferpool CST;
alter tablespace TS_CUSTOMER_25 bufferpool CST;
alter tablespace TS_CUSTOMER_26 bufferpool CST;
alter tablespace TS_CUSTOMER_27 bufferpool CST;
alter tablespace TS_CUSTOMER_28 bufferpool CST;

alter tablespace IS_CUSTOMER_01 bufferpool CST_I;
alter tablespace IS_CUSTOMER_02 bufferpool CST_I;
alter tablespace IS_CUSTOMER_03 bufferpool CST_I;
alter tablespace IS_CUSTOMER_04 bufferpool CST_I;
alter tablespace IS_CUSTOMER_05 bufferpool CST_I;
alter tablespace IS_CUSTOMER_06 bufferpool CST_I;
alter tablespace IS_CUSTOMER_07 bufferpool CST_I;
alter tablespace IS_CUSTOMER_08 bufferpool CST_I;
alter tablespace IS_CUSTOMER_09 bufferpool CST_I;
alter tablespace IS_CUSTOMER_10 bufferpool CST_I;
alter tablespace IS_CUSTOMER_11 bufferpool CST_I;
alter tablespace IS_CUSTOMER_12 bufferpool CST_I;
alter tablespace IS_CUSTOMER_13 bufferpool CST_I;
alter tablespace IS_CUSTOMER_14 bufferpool CST_I;
alter tablespace IS_CUSTOMER_15 bufferpool CST_I;
alter tablespace IS_CUSTOMER_16 bufferpool CST_I;
alter tablespace IS_CUSTOMER_17 bufferpool CST_I;
alter tablespace IS_CUSTOMER_18 bufferpool CST_I;
alter tablespace IS_CUSTOMER_19 bufferpool CST_I;
alter tablespace IS_CUSTOMER_20 bufferpool CST_I;
alter tablespace IS_CUSTOMER_21 bufferpool CST_I;
alter tablespace IS_CUSTOMER_22 bufferpool CST_I;
alter tablespace IS_CUSTOMER_23 bufferpool CST_I;
alter tablespace IS_CUSTOMER_24 bufferpool CST_I;
alter tablespace IS_CUSTOMER_25 bufferpool CST_I;
alter tablespace IS_CUSTOMER_26 bufferpool CST_I;
alter tablespace IS_CUSTOMER_27 bufferpool CST_I;
alter tablespace IS_CUSTOMER_28 bufferpool CST_I;

alter tablespace TS_STOCK_01 bufferpool STK;
alter tablespace TS_STOCK_02 bufferpool STK;
alter tablespace TS_STOCK_03 bufferpool STK;
alter tablespace TS_STOCK_04 bufferpool STK;
alter tablespace TS_STOCK_05 bufferpool STK;
alter tablespace TS_STOCK_06 bufferpool STK;
alter tablespace TS_STOCK_07 bufferpool STK;
alter tablespace TS_STOCK_08 bufferpool STK;
alter tablespace TS_STOCK_09 bufferpool STK;
alter tablespace TS_STOCK_10 bufferpool STK;
alter tablespace TS_STOCK_11 bufferpool STK;
alter tablespace TS_STOCK_12 bufferpool STK;
alter tablespace TS_STOCK_13 bufferpool STK;
alter tablespace TS_STOCK_14 bufferpool STK;
alter tablespace TS_STOCK_15 bufferpool STK;
alter tablespace TS_STOCK_16 bufferpool STK;
alter tablespace TS_STOCK_17 bufferpool STK;
alter tablespace TS_STOCK_18 bufferpool STK;
alter tablespace TS_STOCK_19 bufferpool STK;
alter tablespace TS_STOCK_20 bufferpool STK;
alter tablespace TS_STOCK_21 bufferpool STK;
alter tablespace TS_STOCK_22 bufferpool STK;
alter tablespace TS_STOCK_23 bufferpool STK;
alter tablespace TS_STOCK_24 bufferpool STK;
alter tablespace TS_STOCK_25 bufferpool STK;
alter tablespace TS_STOCK_26 bufferpool STK;
alter tablespace TS_STOCK_27 bufferpool STK;
alter tablespace TS_STOCK_28 bufferpool STK;

connect reset;

terminate;
```

create_bufferpool.ddl

```
-------------------------------------------------------------------------------
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
-------------------------------------------------------------------------------

-- Create Bufferpools
connect to tpcc;
create bufferpool IBMDEFAULT8K size 1000 pagesize 8192;
create bufferpool IBMDEFAULT16K size 1000 pagesize 16384;

create bufferpool WAR   size 1000 pagesize 4096;
create bufferpool DIS   size 1000 pagesize 4096;
create bufferpool ITM   size 1000 pagesize 8192;
```

```
create bufferpool HST   size 1000 pagesize 16384;
create bufferpool NEW   size 1000 pagesize 4096;
create bufferpool NEW_I size 1000 pagesize 4096;
create bufferpool ORD   size 1000 pagesize 8192;
create bufferpool ORD_I size 1000 pagesize 8192;
create bufferpool OLN   size 1000 pagesize 8192;
create bufferpool CST   size 1000 pagesize 4096;
create bufferpool CST_I size 1000 pagesize 8192;
create bufferpool STK   size 1000 pagesize 4096;

connect reset;

terminate;
```

create_database.ddl

```
------------------------------------------------------------------------------
-- Licensed Materials - Property of IBM
--
-- Governed under the terms of the International
-- License Agreement for Non-Warranted Sample Code.
--
-- (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
-- All Rights Reserved.
--
-- US Government Users Restricted Rights - Use, duplication or
-- disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
------------------------------------------------------------------------------

drop database tpcc;
create database tpcc on /home/tpcc/db/tpccdb1 collate using identity
catalog tablespace

managed by system using ('/home/tpcc/db/db1catalog');
```

create_tablespace.ddl

```
connect to tpcc;
drop bufferpool ibmdefaultbp8K;
create bufferpool ibmdefaultbp8K size 2621440 pagesize 8K;
drop bufferpool ibmdefaultbp16K;
create bufferpool ibmdefaultbp16K size 131072 pagesize 16K;
-- now starting to create tablespace on LV

--now creating TS for ts_ware_01 of Database 1

drop tablespace ts_ware_01;
create regular tablespace ts_ware_01 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F01V1WARE' 128,
                                device '/dev/rD1F01V2WARE' 128,
                                device '/dev/rD1F02V1WARE' 128,
                                device '/dev/rD1F02V2WARE' 128,
                                device '/dev/rD1F03V1WARE' 128,
                                device '/dev/rD1F03V2WARE' 128,
                                device '/dev/rD1F04V1WARE' 128,
                                device '/dev/rD1F04V2WARE' 128,
                                device '/dev/rD1F05V1WARE' 128,
                                device '/dev/rD1F05V2WARE' 128,
                                device '/dev/rD1F06V1WARE' 128,
                                device '/dev/rD1F06V2WARE' 128,
                                device '/dev/rD1F07V1WARE' 128,
                                device '/dev/rD1F07V2WARE' 128,
                                device '/dev/rD1F08V1WARE' 128,
                                device '/dev/rD1F08V2WARE' 128,
                                device '/dev/rD1F09V1WARE' 128,
                                device '/dev/rD1F09V2WARE' 128,
                                device '/dev/rD1F10V1WARE' 128,
                                device '/dev/rD1F10V2WARE' 128,
                                device '/dev/rD1F11V1WARE' 128,
                                device '/dev/rD1F11V2WARE' 128,
                                device '/dev/rD1F12V1WARE' 128,
                                device '/dev/rD1F12V2WARE' 128,
                                device '/dev/rD1F13V1WARE' 128,
                                device '/dev/rD1F13V2WARE' 128,
                                device '/dev/rD1F14V1WARE' 128,
                                device '/dev/rD1F14V2WARE' 128,
                                device '/dev/rD1F15V1WARE' 128,
                                device '/dev/rD1F15V2WARE' 128,
                                device '/dev/rD1F16V1WARE' 128,
                                device '/dev/rD1F16V2WARE' 128,
                                device '/dev/rD1F17V1WARE' 128,
                                device '/dev/rD1F17V2WARE' 128,
                                device '/dev/rD1F18V1WARE' 128,
                                device '/dev/rD1F18V2WARE' 128,
                                device '/dev/rD1F19V1WARE' 128,
                                device '/dev/rD1F19V2WARE' 128,
                                device '/dev/rD1F20V1WARE' 128,
                                device '/dev/rD1F20V2WARE' 128,
                                device '/dev/rD1F21V1WARE' 128,
                                device '/dev/rD1F21V2WARE' 128,
                                device '/dev/rD1F22V1WARE' 128,
                                device '/dev/rD1F22V2WARE' 128,
                                device '/dev/rD1F23V1WARE' 128,
                                device '/dev/rD1F23V2WARE' 128,
                                device '/dev/rD1F24V1WARE' 128,
                                device '/dev/rD1F24V2WARE' 128,
                                device '/dev/rD1F25V1WARE' 128,
                                device '/dev/rD1F25V2WARE' 128,
                                device '/dev/rD1F26V1WARE' 128,
                                device '/dev/rD1F26V2WARE' 128,
                                device '/dev/rD1F27V1WARE' 128,
                                device '/dev/rD1F27V2WARE' 128,
                                device '/dev/rD1F28V1WARE' 128,
                                device '/dev/rD1F28V2WARE' 128
                               )
                extentsize 32
                prefetchsize 512;
commit;

--now creating TS for ts_dist_01 of Database 1

drop tablespace ts_dist_01;
create regular tablespace ts_dist_01 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F01V1DIST' 640,
                                device '/dev/rD1F01V2DIST' 640,
                                device '/dev/rD1F02V1DIST' 640,
                                device '/dev/rD1F02V2DIST' 640,
                                device '/dev/rD1F03V1DIST' 640,
                                device '/dev/rD1F03V2DIST' 640,
                                device '/dev/rD1F04V1DIST' 640,
                                device '/dev/rD1F04V2DIST' 640,
                                device '/dev/rD1F05V1DIST' 640,
                                device '/dev/rD1F05V2DIST' 640,
                                device '/dev/rD1F06V1DIST' 640,
                                device '/dev/rD1F06V2DIST' 640,
                                device '/dev/rD1F07V1DIST' 640,
                                device '/dev/rD1F07V2DIST' 640,
                                device '/dev/rD1F08V1DIST' 640,
                                device '/dev/rD1F08V2DIST' 640,
                                device '/dev/rD1F09V1DIST' 640,
                                device '/dev/rD1F09V2DIST' 640,
                                device '/dev/rD1F10V1DIST' 640,
                                device '/dev/rD1F10V2DIST' 640,
                                device '/dev/rD1F11V1DIST' 640,
                                device '/dev/rD1F11V2DIST' 640,
                                device '/dev/rD1F12V1DIST' 640,
                                device '/dev/rD1F12V2DIST' 640,
                                device '/dev/rD1F13V1DIST' 640,
                                device '/dev/rD1F13V2DIST' 640,
                                device '/dev/rD1F14V1DIST' 640,
                                device '/dev/rD1F14V2DIST' 640,
                                device '/dev/rD1F15V1DIST' 640,
                                device '/dev/rD1F15V2DIST' 640,
                                device '/dev/rD1F16V1DIST' 640,
                                device '/dev/rD1F16V2DIST' 640,
                                device '/dev/rD1F17V1DIST' 640,
                                device '/dev/rD1F17V2DIST' 640,
                                device '/dev/rD1F18V1DIST' 640,
                                device '/dev/rD1F18V2DIST' 640,
                                device '/dev/rD1F19V1DIST' 640,
                                device '/dev/rD1F19V2DIST' 640,
                                device '/dev/rD1F20V1DIST' 640,
                                device '/dev/rD1F20V2DIST' 640,
                                device '/dev/rD1F21V1DIST' 640,
                                device '/dev/rD1F21V2DIST' 640,
                                device '/dev/rD1F22V1DIST' 640,
                                device '/dev/rD1F22V2DIST' 640,
                                device '/dev/rD1F23V1DIST' 640,
                                device '/dev/rD1F23V2DIST' 640,
                                device '/dev/rD1F24V1DIST' 640,
                                device '/dev/rD1F24V2DIST' 640,
                                device '/dev/rD1F25V1DIST' 640,
                                device '/dev/rD1F25V2DIST' 640,
                                device '/dev/rD1F26V1DIST' 640,
                                device '/dev/rD1F26V2DIST' 640,
                                device '/dev/rD1F27V1DIST' 640,
                                device '/dev/rD1F27V2DIST' 640,
                                device '/dev/rD1F28V1DIST' 640,
                                device '/dev/rD1F28V2DIST' 640
                               )
                extentsize 64
                prefetchsize 512;
commit;

--now creating TS for ts_item_01 of Database 1

drop tablespace ts_item_01;
create regular tablespace ts_item_01 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F01V1ITEM' 128,
```

```
                                device '/dev/rD1F01V2ITEM' 128,
                                device '/dev/rD1F02V1ITEM' 128,
                                device '/dev/rD1F02V2ITEM' 128,
                                device '/dev/rD1F03V1ITEM' 128,
                                device '/dev/rD1F03V2ITEM' 128,
                                device '/dev/rD1F04V1ITEM' 128,
                                device '/dev/rD1F04V2ITEM' 128,
                                device '/dev/rD1F05V1ITEM' 128,
                                device '/dev/rD1F05V2ITEM' 128,
                                device '/dev/rD1F06V1ITEM' 128,
                                device '/dev/rD1F06V2ITEM' 128,
                                device '/dev/rD1F07V1ITEM' 128,
                                device '/dev/rD1F07V2ITEM' 128,
                                device '/dev/rD1F08V1ITEM' 128,
                                device '/dev/rD1F08V2ITEM' 128,
                                device '/dev/rD1F09V1ITEM' 128,
                                device '/dev/rD1F09V2ITEM' 128,
                                device '/dev/rD1F10V1ITEM' 128,
                                device '/dev/rD1F10V2ITEM' 128,
                                device '/dev/rD1F11V1ITEM' 128,
                                device '/dev/rD1F11V2ITEM' 128,
                                device '/dev/rD1F12V1ITEM' 128,
                                device '/dev/rD1F12V2ITEM' 128,
                                device '/dev/rD1F13V1ITEM' 128,
                                device '/dev/rD1F13V2ITEM' 128,
                                device '/dev/rD1F14V1ITEM' 128,
                                device '/dev/rD1F14V2ITEM' 128,
                                device '/dev/rD1F15V1ITEM' 128,
                                device '/dev/rD1F15V2ITEM' 128,
                                device '/dev/rD1F16V1ITEM' 128,
                                device '/dev/rD1F16V2ITEM' 128,
                                device '/dev/rD1F17V1ITEM' 128,
                                device '/dev/rD1F17V2ITEM' 128,
                                device '/dev/rD1F18V1ITEM' 128,
                                device '/dev/rD1F18V2ITEM' 128,
                                device '/dev/rD1F19V1ITEM' 128,
                                device '/dev/rD1F19V2ITEM' 128,
                                device '/dev/rD1F20V1ITEM' 128,
                                device '/dev/rD1F20V2ITEM' 128,
                                device '/dev/rD1F21V1ITEM' 128,
                                device '/dev/rD1F21V2ITEM' 128,
                                device '/dev/rD1F22V1ITEM' 128,
                                device '/dev/rD1F22V2ITEM' 128,
                                device '/dev/rD1F23V1ITEM' 128,
                                device '/dev/rD1F23V2ITEM' 128,
                                device '/dev/rD1F24V1ITEM' 128,
                                device '/dev/rD1F24V2ITEM' 128,
                                device '/dev/rD1F25V1ITEM' 128,
                                device '/dev/rD1F25V2ITEM' 128,
                                device '/dev/rD1F26V1ITEM' 128,
                                device '/dev/rD1F26V2ITEM' 128,
                                device '/dev/rD1F27V1ITEM' 128,
                                device '/dev/rD1F27V2ITEM' 128,
                                device '/dev/rD1F28V1ITEM' 128,
                                device '/dev/rD1F28V2ITEM' 128
                               )
                extentsize 32
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_stock_01 of Database 1

drop tablespace ts_stock_01;
create regular tablespace ts_stock_01 pagesize 4K
                 managed by database
                 using
                 (
                                device '/dev/rD1F01V1STK01' 8383480,
                                device '/dev/rD1F01V2STK01' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_02 of Database 1

drop tablespace ts_stock_02;
create regular tablespace ts_stock_02 pagesize 4K
                 managed by database
                 using
                 (
                                device '/dev/rD1F02V1STK02' 8383480,
                                device '/dev/rD1F02V2STK02' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_03 of Database 1

drop tablespace ts_stock_03;
create regular tablespace ts_stock_03 pagesize 4K
                 managed by database
                 using
                 (
                                device '/dev/rD1F03V1STK03' 8383480,
                                device '/dev/rD1F03V2STK03' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_04 of Database 1

drop tablespace ts_stock_04;
create regular tablespace ts_stock_04 pagesize 4K
                 managed by database
                 using
                 (
                                device '/dev/rD1F04V1STK04' 8383480,
                                device '/dev/rD1F04V2STK04' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_05 of Database 1

drop tablespace ts_stock_05;
create regular tablespace ts_stock_05 pagesize 4K
                 managed by database
                 using
                 (
                                device '/dev/rD1F05V1STK05' 8383480,
                                device '/dev/rD1F05V2STK05' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_06 of Database 1

drop tablespace ts_stock_06;
create regular tablespace ts_stock_06 pagesize 4K
                 managed by database
                 using
                 (
                                device '/dev/rD1F06V1STK06' 8383480,
                                device '/dev/rD1F06V2STK06' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_07 of Database 1

drop tablespace ts_stock_07;
create regular tablespace ts_stock_07 pagesize 4K
                 managed by database
                 using
                 (
                                device '/dev/rD1F07V1STK07' 8383480,
                                device '/dev/rD1F07V2STK07' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_08 of Database 1

drop tablespace ts_stock_08;
create regular tablespace ts_stock_08 pagesize 4K
                 managed by database
                 using
                 (
                                device '/dev/rD1F08V1STK08' 8383480,
                                device '/dev/rD1F08V2STK08' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_09 of Database 1

drop tablespace ts_stock_09;
create regular tablespace ts_stock_09 pagesize 4K
                 managed by database
                 using
                 (
                                device '/dev/rD1F09V1STK09' 8383480,
                                device '/dev/rD1F09V2STK09' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_10 of Database 1

drop tablespace ts_stock_10;
create regular tablespace ts_stock_10 pagesize 4K
                 managed by database
                 using
                 (
                                device '/dev/rD1F10V1STK10' 8383480,
                                device '/dev/rD1F10V2STK10' 8383480
                               )
                extentsize 256
```

```
                prefetchsize 512;
commit;

--now creating TS for ts_stock_11 of Database 1

drop tablespace ts_stock_11;
create regular tablespace ts_stock_11 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F11V1STK11' 8383480,
                                device '/dev/rD1F11V2STK11' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_12 of Database 1

drop tablespace ts_stock_12;
create regular tablespace ts_stock_12 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F12V1STK12' 8383480,
                                device '/dev/rD1F12V2STK12' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_13 of Database 1

drop tablespace ts_stock_13;
create regular tablespace ts_stock_13 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F13V1STK13' 8383480,
                                device '/dev/rD1F13V2STK13' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_14 of Database 1

drop tablespace ts_stock_14;
create regular tablespace ts_stock_14 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F14V1STK14' 8383480,
                                device '/dev/rD1F14V2STK14' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_15 of Database 1

drop tablespace ts_stock_15;
create regular tablespace ts_stock_15 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F15V1STK15' 8383480,
                                device '/dev/rD1F15V2STK15' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_16 of Database 1

drop tablespace ts_stock_16;
create regular tablespace ts_stock_16 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F16V1STK16' 8383480,
                                device '/dev/rD1F16V2STK16' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_17 of Database 1

drop tablespace ts_stock_17;
create regular tablespace ts_stock_17 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F17V1STK17' 8383480,
                                device '/dev/rD1F17V2STK17' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_18 of Database 1

drop tablespace ts_stock_18;
create regular tablespace ts_stock_18 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F18V1STK18' 8383480,
                                device '/dev/rD1F18V2STK18' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_19 of Database 1

drop tablespace ts_stock_19;
create regular tablespace ts_stock_19 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F19V1STK19' 8383480,
                                device '/dev/rD1F19V2STK19' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_20 of Database 1

drop tablespace ts_stock_20;
create regular tablespace ts_stock_20 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F20V1STK20' 8383480,
                                device '/dev/rD1F20V2STK20' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_21 of Database 1

drop tablespace ts_stock_21;
create regular tablespace ts_stock_21 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F21V1STK21' 8383480,
                                device '/dev/rD1F21V2STK21' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_22 of Database 1

drop tablespace ts_stock_22;
create regular tablespace ts_stock_22 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F22V1STK22' 8383480,
                                device '/dev/rD1F22V2STK22' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_23 of Database 1

drop tablespace ts_stock_23;
create regular tablespace ts_stock_23 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F23V1STK23' 8383480,
                                device '/dev/rD1F23V2STK23' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_stock_24 of Database 1

drop tablespace ts_stock_24;
create regular tablespace ts_stock_24 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F24V1STK24' 8383480,
                                device '/dev/rD1F24V2STK24' 8383480
                               )
                extentsize 256
                prefetchsize 512;
commit;
```

```
--now creating TS for ts_stock_25 of Database 1

drop tablespace ts_stock_25;
create regular tablespace ts_stock_25 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F25V1STK25' 8383480,
                                device '/dev/rD1F25V2STK25' 8383480
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_stock_26 of Database 1

drop tablespace ts_stock_26;
create regular tablespace ts_stock_26 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F26V1STK26' 8383480,
                                device '/dev/rD1F26V2STK26' 8383480
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_stock_27 of Database 1

drop tablespace ts_stock_27;
create regular tablespace ts_stock_27 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F27V1STK27' 8383480,
                                device '/dev/rD1F27V2STK27' 8383480
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_stock_28 of Database 1

drop tablespace ts_stock_28;
create regular tablespace ts_stock_28 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F28V1STK28' 8383480,
                                device '/dev/rD1F28V2STK28' 8383480
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_01 of Database 1

drop tablespace ts_customer_01;
create regular tablespace ts_customer_01 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F01V1CST01' 6066432,
                                device '/dev/rD1F01V2CST01' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_02 of Database 1

drop tablespace ts_customer_02;
create regular tablespace ts_customer_02 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F02V1CST02' 6066432,
                                device '/dev/rD1F02V2CST02' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_03 of Database 1

drop tablespace ts_customer_03;
create regular tablespace ts_customer_03 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F03V1CST03' 6066432,
                                device '/dev/rD1F03V2CST03' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_04 of Database 1

drop tablespace ts_customer_04;
create regular tablespace ts_customer_04 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F04V1CST04' 6066432,
                                device '/dev/rD1F04V2CST04' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_05 of Database 1

drop tablespace ts_customer_05;
create regular tablespace ts_customer_05 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F05V1CST05' 6066432,
                                device '/dev/rD1F05V2CST05' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_06 of Database 1

drop tablespace ts_customer_06;
create regular tablespace ts_customer_06 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F06V1CST06' 6066432,
                                device '/dev/rD1F06V2CST06' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_07 of Database 1

drop tablespace ts_customer_07;
create regular tablespace ts_customer_07 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F07V1CST07' 6066432,
                                device '/dev/rD1F07V2CST07' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_08 of Database 1

drop tablespace ts_customer_08;
create regular tablespace ts_customer_08 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F08V1CST08' 6066432,
                                device '/dev/rD1F08V2CST08' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_09 of Database 1

drop tablespace ts_customer_09;
create regular tablespace ts_customer_09 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F09V1CST09' 6066432,
                                device '/dev/rD1F09V2CST09' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_10 of Database 1

drop tablespace ts_customer_10;
create regular tablespace ts_customer_10 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F10V1CST10' 6066432,
                                device '/dev/rD1F10V2CST10' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_11 of Database 1
```

```
drop tablespace ts_customer_11;
create regular tablespace ts_customer_11 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F11V1CST11' 6066432,
                                device '/dev/rD1F11V2CST11' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_12 of Database 1

drop tablespace ts_customer_12;
create regular tablespace ts_customer_12 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F12V1CST12' 6066432,
                                device '/dev/rD1F12V2CST12' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_13 of Database 1

drop tablespace ts_customer_13;
create regular tablespace ts_customer_13 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F13V1CST13' 6066432,
                                device '/dev/rD1F13V2CST13' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_14 of Database 1

drop tablespace ts_customer_14;
create regular tablespace ts_customer_14 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F14V1CST14' 6066432,
                                device '/dev/rD1F14V2CST14' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_15 of Database 1

drop tablespace ts_customer_15;
create regular tablespace ts_customer_15 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F15V1CST15' 6066432,
                                device '/dev/rD1F15V2CST15' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_16 of Database 1

drop tablespace ts_customer_16;
create regular tablespace ts_customer_16 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F16V1CST16' 6066432,
                                device '/dev/rD1F16V2CST16' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_17 of Database 1

drop tablespace ts_customer_17;
create regular tablespace ts_customer_17 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F17V1CST17' 6066432,
                                device '/dev/rD1F17V2CST17' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_18 of Database 1

drop tablespace ts_customer_18;
create regular tablespace ts_customer_18 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F18V1CST18' 6066432,
                                device '/dev/rD1F18V2CST18' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_19 of Database 1

drop tablespace ts_customer_19;
create regular tablespace ts_customer_19 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F19V1CST19' 6066432,
                                device '/dev/rD1F19V2CST19' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_20 of Database 1

drop tablespace ts_customer_20;
create regular tablespace ts_customer_20 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F20V1CST20' 6066432,
                                device '/dev/rD1F20V2CST20' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_21 of Database 1

drop tablespace ts_customer_21;
create regular tablespace ts_customer_21 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F21V1CST21' 6066432,
                                device '/dev/rD1F21V2CST21' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_22 of Database 1

drop tablespace ts_customer_22;
create regular tablespace ts_customer_22 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F22V1CST22' 6066432,
                                device '/dev/rD1F22V2CST22' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_23 of Database 1

drop tablespace ts_customer_23;
create regular tablespace ts_customer_23 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F23V1CST23' 6066432,
                                device '/dev/rD1F23V2CST23' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_24 of Database 1

drop tablespace ts_customer_24;
create regular tablespace ts_customer_24 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F24V1CST24' 6066432,
                                device '/dev/rD1F24V2CST24' 6066432
                               )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for ts_customer_25 of Database 1

drop tablespace ts_customer_25;
```

```
create regular tablespace ts_customer_25 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F25V1CST25' 6066432,
                                device '/dev/rD1F25V2CST25' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_26 of Database 1

drop tablespace ts_customer_26;
create regular tablespace ts_customer_26 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F26V1CST26' 6066432,
                                device '/dev/rD1F26V2CST26' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_27 of Database 1

drop tablespace ts_customer_27;
create regular tablespace ts_customer_27 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F27V1CST27' 6066432,
                                device '/dev/rD1F27V2CST27' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for ts_customer_28 of Database 1

drop tablespace ts_customer_28;
create regular tablespace ts_customer_28 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F28V1CST28' 6066432,
                                device '/dev/rD1F28V2CST28' 6066432
                               )
               extentsize 256
               prefetchsize 512;
commit;

--now creating TS for is_customer_01 of Database 1

drop tablespace is_customer_01;
create regular tablespace is_customer_01 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F01V1CSTI01' 221952,
                                device '/dev/rD1F01V2CSTI01' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_02 of Database 1

drop tablespace is_customer_02;
create regular tablespace is_customer_02 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F02V1CSTI02' 221952,
                                device '/dev/rD1F02V2CSTI02' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_03 of Database 1

drop tablespace is_customer_03;
create regular tablespace is_customer_03 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F03V1CSTI03' 221952,
                                device '/dev/rD1F03V2CSTI03' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_04 of Database 1

drop tablespace is_customer_04;
create regular tablespace is_customer_04 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F04V1CSTI04' 221952,
                                device '/dev/rD1F04V2CSTI04' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_05 of Database 1

drop tablespace is_customer_05;
create regular tablespace is_customer_05 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F05V1CSTI05' 221952,
                                device '/dev/rD1F05V2CSTI05' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_06 of Database 1

drop tablespace is_customer_06;
create regular tablespace is_customer_06 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F06V1CSTI06' 221952,
                                device '/dev/rD1F06V2CSTI06' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_07 of Database 1

drop tablespace is_customer_07;
create regular tablespace is_customer_07 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F07V1CSTI07' 221952,
                                device '/dev/rD1F07V2CSTI07' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_08 of Database 1

drop tablespace is_customer_08;
create regular tablespace is_customer_08 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F08V1CSTI08' 221952,
                                device '/dev/rD1F08V2CSTI08' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_09 of Database 1

drop tablespace is_customer_09;
create regular tablespace is_customer_09 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F09V1CSTI09' 221952,
                                device '/dev/rD1F09V2CSTI09' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_10 of Database 1

drop tablespace is_customer_10;
create regular tablespace is_customer_10 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F10V1CSTI10' 221952,
                                device '/dev/rD1F10V2CSTI10' 221952
                               )
```

```
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_11 of Database 1

drop tablespace is_customer_11;
create regular tablespace is_customer_11 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F11V1CSTI11' 221952,
                                device '/dev/rD1F11V2CSTI11' 221952
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_12 of Database 1

drop tablespace is_customer_12;
create regular tablespace is_customer_12 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F12V1CSTI12' 221952,
                                device '/dev/rD1F12V2CSTI12' 221952
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_13 of Database 1

drop tablespace is_customer_13;
create regular tablespace is_customer_13 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F13V1CSTI13' 221952,
                                device '/dev/rD1F13V2CSTI13' 221952
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_14 of Database 1

drop tablespace is_customer_14;
create regular tablespace is_customer_14 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F14V1CSTI14' 221952,
                                device '/dev/rD1F14V2CSTI14' 221952
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_15 of Database 1

drop tablespace is_customer_15;
create regular tablespace is_customer_15 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F15V1CSTI15' 221952,
                                device '/dev/rD1F15V2CSTI15' 221952
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_16 of Database 1

drop tablespace is_customer_16;
create regular tablespace is_customer_16 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F16V1CSTI16' 221952,
                                device '/dev/rD1F16V2CSTI16' 221952
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_17 of Database 1

drop tablespace is_customer_17;
create regular tablespace is_customer_17 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F17V1CSTI17' 221952,
                                device '/dev/rD1F17V2CSTI17' 221952
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_18 of Database 1

drop tablespace is_customer_18;
create regular tablespace is_customer_18 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F18V1CSTI18' 221952,
                                device '/dev/rD1F18V2CSTI18' 221952
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_19 of Database 1

drop tablespace is_customer_19;
create regular tablespace is_customer_19 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F19V1CSTI19' 221952,
                                device '/dev/rD1F19V2CSTI19' 221952
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_20 of Database 1

drop tablespace is_customer_20;
create regular tablespace is_customer_20 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F20V1CSTI20' 221952,
                                device '/dev/rD1F20V2CSTI20' 221952
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_21 of Database 1

drop tablespace is_customer_21;
create regular tablespace is_customer_21 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F21V1CSTI21' 221952,
                                device '/dev/rD1F21V2CSTI21' 221952
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_22 of Database 1

drop tablespace is_customer_22;
create regular tablespace is_customer_22 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F22V1CSTI22' 221952,
                                device '/dev/rD1F22V2CSTI22' 221952
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_23 of Database 1

drop tablespace is_customer_23;
create regular tablespace is_customer_23 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F23V1CSTI23' 221952,
                                device '/dev/rD1F23V2CSTI23' 221952
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
```

```
commit;

--now creating TS for is_customer_24 of Database 1

drop tablespace is_customer_24;
create regular tablespace is_customer_24 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F24V1CSTI24' 221952,
                                device '/dev/rD1F24V2CSTI24' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_25 of Database 1

drop tablespace is_customer_25;
create regular tablespace is_customer_25 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F25V1CSTI25' 221952,
                                device '/dev/rD1F25V2CSTI25' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_26 of Database 1

drop tablespace is_customer_26;
create regular tablespace is_customer_26 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F26V1CSTI26' 221952,
                                device '/dev/rD1F26V2CSTI26' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_27 of Database 1

drop tablespace is_customer_27;
create regular tablespace is_customer_27 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F27V1CSTI27' 221952,
                                device '/dev/rD1F27V2CSTI27' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_customer_28 of Database 1

drop tablespace is_customer_28;
create regular tablespace is_customer_28 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F28V1CSTI28' 221952,
                                device '/dev/rD1F28V2CSTI28' 221952
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_history_01 of Database 1

drop tablespace ts_history_01;
create regular tablespace ts_history_01 pagesize 16K
                managed by database
                using
                (
                                device '/dev/rD1F01V1HIST' 177920,
                                device '/dev/rD1F01V2HIST' 177920,
                                device '/dev/rD1F02V1HIST' 177920,
                                device '/dev/rD1F02V2HIST' 177920,
                                device '/dev/rD1F03V1HIST' 177920,
                                device '/dev/rD1F03V2HIST' 177920,
                                device '/dev/rD1F04V1HIST' 177920,
                                device '/dev/rD1F04V2HIST' 177920,
                                device '/dev/rD1F05V1HIST' 177920,
                                device '/dev/rD1F05V2HIST' 177920,
                                device '/dev/rD1F06V1HIST' 177920,
                                device '/dev/rD1F06V2HIST' 177920,
                                device '/dev/rD1F07V1HIST' 177920,
                                device '/dev/rD1F07V2HIST' 177920,
                                device '/dev/rD1F08V1HIST' 177920,
                                device '/dev/rD1F08V2HIST' 177920,
                                device '/dev/rD1F09V1HIST' 177920,
                                device '/dev/rD1F09V2HIST' 177920,
                                device '/dev/rD1F10V1HIST' 177920,
                                device '/dev/rD1F10V2HIST' 177920,
                                device '/dev/rD1F11V1HIST' 177920,
                                device '/dev/rD1F11V2HIST' 177920,
                                device '/dev/rD1F12V1HIST' 177920,
                                device '/dev/rD1F12V2HIST' 177920,
                                device '/dev/rD1F13V1HIST' 177920,
                                device '/dev/rD1F13V2HIST' 177920,
                                device '/dev/rD1F14V1HIST' 177920,
                                device '/dev/rD1F14V2HIST' 177920,
                                device '/dev/rD1F15V1HIST' 177920,
                                device '/dev/rD1F15V2HIST' 177920,
                                device '/dev/rD1F16V1HIST' 177920,
                                device '/dev/rD1F16V2HIST' 177920,
                                device '/dev/rD1F17V1HIST' 177920,
                                device '/dev/rD1F17V2HIST' 177920,
                                device '/dev/rD1F18V1HIST' 177920,
                                device '/dev/rD1F18V2HIST' 177920,
                                device '/dev/rD1F19V1HIST' 177920,
                                device '/dev/rD1F19V2HIST' 177920,
                                device '/dev/rD1F20V1HIST' 177920,
                                device '/dev/rD1F20V2HIST' 177920,
                                device '/dev/rD1F21V1HIST' 177920,
                                device '/dev/rD1F21V2HIST' 177920,
                                device '/dev/rD1F22V1HIST' 177920,
                                device '/dev/rD1F22V2HIST' 177920,
                                device '/dev/rD1F23V1HIST' 177920,
                                device '/dev/rD1F23V2HIST' 177920,
                                device '/dev/rD1F24V1HIST' 177920,
                                device '/dev/rD1F24V2HIST' 177920,
                                device '/dev/rD1F25V1HIST' 177920,
                                device '/dev/rD1F25V2HIST' 177920,
                                device '/dev/rD1F26V1HIST' 177920,
                                device '/dev/rD1F26V2HIST' 177920,
                                device '/dev/rD1F27V1HIST' 177920,
                                device '/dev/rD1F27V2HIST' 177920,
                                device '/dev/rD1F28V1HIST' 177920,
                                device '/dev/rD1F28V2HIST' 177920
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp16K;
commit;

--now creating TS for ts_order_01 of Database 1

drop tablespace ts_order_01;
create regular tablespace ts_order_01 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F01V1ORD' 187648,
                                device '/dev/rD1F01V2ORD' 187648,
                                device '/dev/rD1F02V1ORD' 187648,
                                device '/dev/rD1F02V2ORD' 187648,
                                device '/dev/rD1F03V1ORD' 187648,
                                device '/dev/rD1F03V2ORD' 187648,
                                device '/dev/rD1F04V1ORD' 187648,
                                device '/dev/rD1F04V2ORD' 187648,
                                device '/dev/rD1F05V1ORD' 187648,
                                device '/dev/rD1F05V2ORD' 187648,
                                device '/dev/rD1F06V1ORD' 187648,
                                device '/dev/rD1F06V2ORD' 187648,
                                device '/dev/rD1F07V1ORD' 187648,
                                device '/dev/rD1F07V2ORD' 187648,
                                device '/dev/rD1F08V1ORD' 187648,
                                device '/dev/rD1F08V2ORD' 187648,
                                device '/dev/rD1F09V1ORD' 187648,
                                device '/dev/rD1F09V2ORD' 187648,
                                device '/dev/rD1F10V1ORD' 187648,
                                device '/dev/rD1F10V2ORD' 187648,
                                device '/dev/rD1F11V1ORD' 187648,
                                device '/dev/rD1F11V2ORD' 187648,
                                device '/dev/rD1F12V1ORD' 187648,
                                device '/dev/rD1F12V2ORD' 187648,
                                device '/dev/rD1F13V1ORD' 187648,
                                device '/dev/rD1F13V2ORD' 187648,
                                device '/dev/rD1F14V1ORD' 187648,
                                device '/dev/rD1F14V2ORD' 187648,
                                device '/dev/rD1F15V1ORD' 187648,
                                device '/dev/rD1F15V2ORD' 187648,
                                device '/dev/rD1F16V1ORD' 187648,
                                device '/dev/rD1F16V2ORD' 187648,
                                device '/dev/rD1F17V1ORD' 187648,
                                device '/dev/rD1F17V2ORD' 187648,
                                device '/dev/rD1F18V1ORD' 187648,
                                device '/dev/rD1F18V2ORD' 187648,
                                device '/dev/rD1F19V1ORD' 187648,
                                device '/dev/rD1F19V2ORD' 187648,
                                device '/dev/rD1F20V1ORD' 187648,
                                device '/dev/rD1F20V2ORD' 187648,
                                device '/dev/rD1F21V1ORD' 187648,
                                device '/dev/rD1F21V2ORD' 187648,
                                device '/dev/rD1F22V1ORD' 187648,
                                device '/dev/rD1F22V2ORD' 187648,
                                device '/dev/rD1F23V1ORD' 187648,
```

```
                                device '/dev/rD1F23V2ORD' 187648,
                                device '/dev/rD1F24V1ORD' 187648,
                                device '/dev/rD1F24V2ORD' 187648,
                                device '/dev/rD1F25V1ORD' 187648,
                                device '/dev/rD1F25V2ORD' 187648,
                                device '/dev/rD1F26V1ORD' 187648,
                                device '/dev/rD1F26V2ORD' 187648,
                                device '/dev/rD1F27V1ORD' 187648,
                                device '/dev/rD1F27V2ORD' 187648,
                                device '/dev/rD1F28V1ORD' 187648,
                                device '/dev/rD1F28V2ORD' 187648
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for is_order_01 of Database 1

drop tablespace is_order_01;
create regular tablespace is_order_01 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F01V1ORDI' 168192,
                                device '/dev/rD1F01V2ORDI' 168192,
                                device '/dev/rD1F02V1ORDI' 168192,
                                device '/dev/rD1F02V2ORDI' 168192,
                                device '/dev/rD1F03V1ORDI' 168192,
                                device '/dev/rD1F03V2ORDI' 168192,
                                device '/dev/rD1F04V1ORDI' 168192,
                                device '/dev/rD1F04V2ORDI' 168192,
                                device '/dev/rD1F05V1ORDI' 168192,
                                device '/dev/rD1F05V2ORDI' 168192,
                                device '/dev/rD1F06V1ORDI' 168192,
                                device '/dev/rD1F06V2ORDI' 168192,
                                device '/dev/rD1F07V1ORDI' 168192,
                                device '/dev/rD1F07V2ORDI' 168192,
                                device '/dev/rD1F08V1ORDI' 168192,
                                device '/dev/rD1F08V2ORDI' 168192,
                                device '/dev/rD1F09V1ORDI' 168192,
                                device '/dev/rD1F09V2ORDI' 168192,
                                device '/dev/rD1F10V1ORDI' 168192,
                                device '/dev/rD1F10V2ORDI' 168192,
                                device '/dev/rD1F11V1ORDI' 168192,
                                device '/dev/rD1F11V2ORDI' 168192,
                                device '/dev/rD1F12V1ORDI' 168192,
                                device '/dev/rD1F12V2ORDI' 168192,
                                device '/dev/rD1F13V1ORDI' 168192,
                                device '/dev/rD1F13V2ORDI' 168192,
                                device '/dev/rD1F14V1ORDI' 168192,
                                device '/dev/rD1F14V2ORDI' 168192,
                                device '/dev/rD1F15V1ORDI' 168192,
                                device '/dev/rD1F15V2ORDI' 168192,
                                device '/dev/rD1F16V1ORDI' 168192,
                                device '/dev/rD1F16V2ORDI' 168192,
                                device '/dev/rD1F17V1ORDI' 168192,
                                device '/dev/rD1F17V2ORDI' 168192,
                                device '/dev/rD1F18V1ORDI' 168192,
                                device '/dev/rD1F18V2ORDI' 168192,
                                device '/dev/rD1F19V1ORDI' 168192,
                                device '/dev/rD1F19V2ORDI' 168192,
                                device '/dev/rD1F20V1ORDI' 168192,
                                device '/dev/rD1F20V2ORDI' 168192,
                                device '/dev/rD1F21V1ORDI' 168192,
                                device '/dev/rD1F21V2ORDI' 168192,
                                device '/dev/rD1F22V1ORDI' 168192,
                                device '/dev/rD1F22V2ORDI' 168192,
                                device '/dev/rD1F23V1ORDI' 168192,
                                device '/dev/rD1F23V2ORDI' 168192,
                                device '/dev/rD1F24V1ORDI' 168192,
                                device '/dev/rD1F24V2ORDI' 168192,
                                device '/dev/rD1F25V1ORDI' 168192,
                                device '/dev/rD1F25V2ORDI' 168192,
                                device '/dev/rD1F26V1ORDI' 168192,
                                device '/dev/rD1F26V2ORDI' 168192,
                                device '/dev/rD1F27V1ORDI' 168192,
                                device '/dev/rD1F27V2ORDI' 168192,
                                device '/dev/rD1F28V1ORDI' 168192,
                                device '/dev/rD1F28V2ORDI' 168192
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_01 of Database 1

drop tablespace ts_orderline_01;
create regular tablespace ts_orderline_01 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F01V1ORL01' 5075456,
                                device '/dev/rD1F01V2ORL01' 5075456
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_02 of Database 1

drop tablespace ts_orderline_02;
create regular tablespace ts_orderline_02 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F02V1ORL02' 5075456,
                                device '/dev/rD1F02V2ORL02' 5075456
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_03 of Database 1

drop tablespace ts_orderline_03;
create regular tablespace ts_orderline_03 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F03V1ORL03' 5075456,
                                device '/dev/rD1F03V2ORL03' 5075456
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_04 of Database 1

drop tablespace ts_orderline_04;
create regular tablespace ts_orderline_04 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F04V1ORL04' 5075456,
                                device '/dev/rD1F04V2ORL04' 5075456
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_05 of Database 1

drop tablespace ts_orderline_05;
create regular tablespace ts_orderline_05 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F05V1ORL05' 5075456,
                                device '/dev/rD1F05V2ORL05' 5075456
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_06 of Database 1

drop tablespace ts_orderline_06;
create regular tablespace ts_orderline_06 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F06V1ORL06' 5075456,
                                device '/dev/rD1F06V2ORL06' 5075456
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_07 of Database 1

drop tablespace ts_orderline_07;
create regular tablespace ts_orderline_07 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F07V1ORL07' 5075456,
                                device '/dev/rD1F07V2ORL07' 5075456
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_08 of Database 1

drop tablespace ts_orderline_08;
create regular tablespace ts_orderline_08 pagesize 8K
                 managed by database
                 using
                (
```

```
                                device '/dev/rD1F08V1ORL08' 5075456,
                                device '/dev/rD1F08V2ORL08' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_09 of Database 1

drop tablespace ts_orderline_09;
create regular tablespace ts_orderline_09 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F09V1ORL09' 5075456,
                                device '/dev/rD1F09V2ORL09' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_10 of Database 1

drop tablespace ts_orderline_10;
create regular tablespace ts_orderline_10 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F10V1ORL10' 5075456,
                                device '/dev/rD1F10V2ORL10' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_11 of Database 1

drop tablespace ts_orderline_11;
create regular tablespace ts_orderline_11 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F11V1ORL11' 5075456,
                                device '/dev/rD1F11V2ORL11' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_12 of Database 1

drop tablespace ts_orderline_12;
create regular tablespace ts_orderline_12 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F12V1ORL12' 5075456,
                                device '/dev/rD1F12V2ORL12' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_13 of Database 1

drop tablespace ts_orderline_13;
create regular tablespace ts_orderline_13 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F13V1ORL13' 5075456,
                                device '/dev/rD1F13V2ORL13' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_14 of Database 1

drop tablespace ts_orderline_14;
create regular tablespace ts_orderline_14 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F14V1ORL14' 5075456,
                                device '/dev/rD1F14V2ORL14' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_15 of Database 1

drop tablespace ts_orderline_15;
create regular tablespace ts_orderline_15 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F15V1ORL15' 5075456,
                                device '/dev/rD1F15V2ORL15' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_16 of Database 1

drop tablespace ts_orderline_16;
create regular tablespace ts_orderline_16 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F16V1ORL16' 5075456,
                                device '/dev/rD1F16V2ORL16' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_17 of Database 1

drop tablespace ts_orderline_17;
create regular tablespace ts_orderline_17 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F17V1ORL17' 5075456,
                                device '/dev/rD1F17V2ORL17' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_18 of Database 1

drop tablespace ts_orderline_18;
create regular tablespace ts_orderline_18 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F18V1ORL18' 5075456,
                                device '/dev/rD1F18V2ORL18' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_19 of Database 1

drop tablespace ts_orderline_19;
create regular tablespace ts_orderline_19 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F19V1ORL19' 5075456,
                                device '/dev/rD1F19V2ORL19' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_20 of Database 1

drop tablespace ts_orderline_20;
create regular tablespace ts_orderline_20 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F20V1ORL20' 5075456,
                                device '/dev/rD1F20V2ORL20' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_21 of Database 1

drop tablespace ts_orderline_21;
create regular tablespace ts_orderline_21 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F21V1ORL21' 5075456,
                                device '/dev/rD1F21V2ORL21' 5075456
                                )
```

```
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_22 of Database 1

drop tablespace ts_orderline_22;
create regular tablespace ts_orderline_22 pagesize 8K
                 managed by database
                 using
                 (
                                 device '/dev/rD1F22V1ORL22' 5075456,
                                 device '/dev/rD1F22V2ORL22' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_23 of Database 1

drop tablespace ts_orderline_23;
create regular tablespace ts_orderline_23 pagesize 8K
                 managed by database
                 using
                 (
                                 device '/dev/rD1F23V1ORL23' 5075456,
                                 device '/dev/rD1F23V2ORL23' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_24 of Database 1

drop tablespace ts_orderline_24;
create regular tablespace ts_orderline_24 pagesize 8K
                 managed by database
                 using
                 (
                                 device '/dev/rD1F24V1ORL24' 5075456,
                                 device '/dev/rD1F24V2ORL24' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_25 of Database 1

drop tablespace ts_orderline_25;
create regular tablespace ts_orderline_25 pagesize 8K
                 managed by database
                 using
                 (
                                 device '/dev/rD1F25V1ORL25' 5075456,
                                 device '/dev/rD1F25V2ORL25' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_26 of Database 1

drop tablespace ts_orderline_26;
create regular tablespace ts_orderline_26 pagesize 8K
                 managed by database
                 using
                 (
                                 device '/dev/rD1F26V1ORL26' 5075456,
                                 device '/dev/rD1F26V2ORL26' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_27 of Database 1

drop tablespace ts_orderline_27;
create regular tablespace ts_orderline_27 pagesize 8K
                 managed by database
                 using
                 (
                                 device '/dev/rD1F27V1ORL27' 5075456,
                                 device '/dev/rD1F27V2ORL27' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_orderline_28 of Database 1

drop tablespace ts_orderline_28;
create regular tablespace ts_orderline_28 pagesize 8K
                 managed by database
                 using
                 (
                                 device '/dev/rD1F28V1ORL28' 5075456,
                                 device '/dev/rD1F28V2ORL28' 5075456
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

--now creating TS for ts_neword_01 of Database 1

drop tablespace ts_neword_01;
create regular tablespace ts_neword_01 pagesize 4K
                 managed by database
                 using
                 (
                                 device '/dev/rD1F01V1NOR' 109568,
                                 device '/dev/rD1F01V2NOR' 109568,
                                 device '/dev/rD1F02V1NOR' 109568,
                                 device '/dev/rD1F02V2NOR' 109568,
                                 device '/dev/rD1F03V1NOR' 109568,
                                 device '/dev/rD1F03V2NOR' 109568,
                                 device '/dev/rD1F04V1NOR' 109568,
                                 device '/dev/rD1F04V2NOR' 109568,
                                 device '/dev/rD1F05V1NOR' 109568,
                                 device '/dev/rD1F05V2NOR' 109568,
                                 device '/dev/rD1F06V1NOR' 109568,
                                 device '/dev/rD1F06V2NOR' 109568,
                                 device '/dev/rD1F07V1NOR' 109568,
                                 device '/dev/rD1F07V2NOR' 109568,
                                 device '/dev/rD1F08V1NOR' 109568,
                                 device '/dev/rD1F08V2NOR' 109568,
                                 device '/dev/rD1F09V1NOR' 109568,
                                 device '/dev/rD1F09V2NOR' 109568,
                                 device '/dev/rD1F10V1NOR' 109568,
                                 device '/dev/rD1F10V2NOR' 109568,
                                 device '/dev/rD1F11V1NOR' 109568,
                                 device '/dev/rD1F11V2NOR' 109568,
                                 device '/dev/rD1F12V1NOR' 109568,
                                 device '/dev/rD1F12V2NOR' 109568,
                                 device '/dev/rD1F13V1NOR' 109568,
                                 device '/dev/rD1F13V2NOR' 109568,
                                 device '/dev/rD1F14V1NOR' 109568,
                                 device '/dev/rD1F14V2NOR' 109568,
                                 device '/dev/rD1F15V1NOR' 109568,
                                 device '/dev/rD1F15V2NOR' 109568,
                                 device '/dev/rD1F16V1NOR' 109568,
                                 device '/dev/rD1F16V2NOR' 109568,
                                 device '/dev/rD1F17V1NOR' 109568,
                                 device '/dev/rD1F17V2NOR' 109568,
                                 device '/dev/rD1F18V1NOR' 109568,
                                 device '/dev/rD1F18V2NOR' 109568,
                                 device '/dev/rD1F19V1NOR' 109568,
                                 device '/dev/rD1F19V2NOR' 109568,
                                 device '/dev/rD1F20V1NOR' 109568,
                                 device '/dev/rD1F20V2NOR' 109568,
                                 device '/dev/rD1F21V1NOR' 109568,
                                 device '/dev/rD1F21V2NOR' 109568,
                                 device '/dev/rD1F22V1NOR' 109568,
                                 device '/dev/rD1F22V2NOR' 109568,
                                 device '/dev/rD1F23V1NOR' 109568,
                                 device '/dev/rD1F23V2NOR' 109568,
                                 device '/dev/rD1F24V1NOR' 109568,
                                 device '/dev/rD1F24V2NOR' 109568,
                                 device '/dev/rD1F25V1NOR' 109568,
                                 device '/dev/rD1F25V2NOR' 109568,
                                 device '/dev/rD1F26V1NOR' 109568,
                                 device '/dev/rD1F26V2NOR' 109568,
                                 device '/dev/rD1F27V1NOR' 109568,
                                 device '/dev/rD1F27V2NOR' 109568,
                                 device '/dev/rD1F28V1NOR' 109568,
                                 device '/dev/rD1F28V2NOR' 109568
                                )
                extentsize 256
                prefetchsize 512;
commit;

--now creating TS for is_neword_01 of Database 1

drop tablespace is_neword_01;
create regular tablespace is_neword_01 pagesize 4K
                 managed by database
                 using
                 (
                                 device '/dev/rD1F01V1NORI' 109568,
                                 device '/dev/rD1F01V2NORI' 109568,
                                 device '/dev/rD1F02V1NORI' 109568,
                                 device '/dev/rD1F02V2NORI' 109568,
                                 device '/dev/rD1F03V1NORI' 109568,
                                 device '/dev/rD1F03V2NORI' 109568,
                                 device '/dev/rD1F04V1NORI' 109568,
                                 device '/dev/rD1F04V2NORI' 109568,
                                 device '/dev/rD1F05V1NORI' 109568,
                                 device '/dev/rD1F05V2NORI' 109568,
                                 device '/dev/rD1F06V1NORI' 109568,
                                 device '/dev/rD1F06V2NORI' 109568,
                                 device '/dev/rD1F07V1NORI' 109568,
```

```
                                device '/dev/rD1F07V2NORI' 109568,
                                device '/dev/rD1F08V1NORI' 109568,
                                device '/dev/rD1F08V2NORI' 109568,
                                device '/dev/rD1F09V1NORI' 109568,
                                device '/dev/rD1F09V2NORI' 109568,
                                device '/dev/rD1F10V1NORI' 109568,
                                device '/dev/rD1F10V2NORI' 109568,
                                device '/dev/rD1F11V1NORI' 109568,
                                device '/dev/rD1F11V2NORI' 109568,
                                device '/dev/rD1F12V1NORI' 109568,
                                device '/dev/rD1F12V2NORI' 109568,
                                device '/dev/rD1F13V1NORI' 109568,
                                device '/dev/rD1F13V2NORI' 109568,
                                device '/dev/rD1F14V1NORI' 109568,
                                device '/dev/rD1F14V2NORI' 109568,
                                device '/dev/rD1F15V1NORI' 109568,
                                device '/dev/rD1F15V2NORI' 109568,
                                device '/dev/rD1F16V1NORI' 109568,
                                device '/dev/rD1F16V2NORI' 109568,
                                device '/dev/rD1F17V1NORI' 109568,
                                device '/dev/rD1F17V2NORI' 109568,
                                device '/dev/rD1F18V1NORI' 109568,
                                device '/dev/rD1F18V2NORI' 109568,
                                device '/dev/rD1F19V1NORI' 109568,
                                device '/dev/rD1F19V2NORI' 109568,
                                device '/dev/rD1F20V1NORI' 109568,
                                device '/dev/rD1F20V2NORI' 109568,
                                device '/dev/rD1F21V1NORI' 109568,
                                device '/dev/rD1F21V2NORI' 109568,
                                device '/dev/rD1F22V1NORI' 109568,
                                device '/dev/rD1F22V2NORI' 109568,
                                device '/dev/rD1F23V1NORI' 109568,
                                device '/dev/rD1F23V2NORI' 109568,
                                device '/dev/rD1F24V1NORI' 109568,
                                device '/dev/rD1F24V2NORI' 109568,
                                device '/dev/rD1F25V1NORI' 109568,
                                device '/dev/rD1F25V2NORI' 109568,
                                device '/dev/rD1F26V1NORI' 109568,
                                device '/dev/rD1F26V2NORI' 109568,
                                device '/dev/rD1F27V1NORI' 109568,
                                device '/dev/rD1F27V2NORI' 109568,
                                device '/dev/rD1F28V1NORI' 109568,
                                device '/dev/rD1F28V2NORI' 109568
                               )
                extentsize 256
                prefetchsize 512;
commit;

alter bufferpool ibmdefaultbp DEFERRED size 5000000;

connect reset;
```

C.2 Data Generation Code

Makefile

```
#################################################################################
## Licensed Materials - Property of IBM
##
## Governed under the terms of the International
## License Agreement for Non-Warranted Sample Code.
##
## (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
## All Rights Reserved.
##
## US Government Users Restricted Rights - Use, duplication or
## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#################################################################################

#
# Makefile - Build gendata tool
#

include $(TPCC_ROOT)/Makefile.config

# ###############################################################################
# Preprocessor, Compiler and Linker Flags
# ###############################################################################

INCLUDE =       -I$(TPCC_SQLLIB)/include -I$(TPCC_ROOT)/include

CFLAGS =        $(INCLUDE) $(CFLAGS_OS) -DLINT_ARGS -DSQLA_NOLINES \
                        -D$(DB2EDITION) -D$(DB2VERSION) $(CFLAGS_DEBUG)

LDFLAGS =   $(LDFLAGS_EXEC) $(LDFLAGS_LIB)

# ###############################################################################
# File Collections
# ###############################################################################

OBJS =                  tpccrnd$(OBJEXT) \
                        $(TPCC_ROOT)/Src.Common/tpccmisc$(OBJEXT)
OBJ_EEE =   $(TPCC_ROOT)/Src.Common/tpcclwh$(OBJEXT)

EXEC =                  gendata$(BINEXT)

# ###############################################################################
# End-User Targets
# ###############################################################################

all:            $(EXEC)

clean:
                - $(ERASE) *$(OBJEXT) $(EXEC)

# ###############################################################################
# Build Rules
# ###############################################################################

.SUFFIXES:
.SUFFIXES:      $(OBJEXT) .c

$(EXEC):
                $(LD_EXEC) $(LDFLAGS) $(OBJS) $*$(OBJEXT) $(LDFLAGS_OUT)$@

# ###############################################################################
# Dependencies
# ###############################################################################

# Link Dependencies
gendata$(BINEXT):       $(OBJS) gendata$(OBJEXT)

# Build Dependencies
# Source
gendata$(OBJEXT):       gendata.c

# Headers

gendata.c:      $(TPCC_ROOT)/include/tpccrnd.h $(TPCC_ROOT)/include/lval.h
```

db2tpcc.h

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * db2tpcc.h - Macros and Miscellany
 */

#ifndef __DB2TPCC_H
#define __DB2TPCC_H

#include <sys/types.h>

#include "lval.h"

/* ************************************************************************ */
/* Transaction Return Codes (s_transtatus)                                  */
/* ************************************************************************ */

#define INVALID_ITEM          100
#define TRAN_OK               0
#define FATAL_SQLERROR        -1

/* ************************************************************************ */
/* Definition of Unused and Bad Items                                       */
/* ************************************************************************ */
/* Define unused item ID to be 0. This allows the SUT to determine the      */
/* number of items in the order as required by 2.4.1.3 and 2.4.2.2 since    */
/* the assumption that any item with OL_I_ID = 0 is unused will be true.    */
/* This in turn requires that the value used for an invalid item is         */
/* equal to ITEMS + 1.                                                      */
/* ************************************************************************ */

#define INVALID_ITEM_ID (2 * ITEMS) + 1
#define UNUSED_ITEM_ID 0

#define MIN_WAREHOUSE 1
#define MAX_WAREHOUSE WAREHOUSES

/*****************************************************************************/
/* NURand Constants                                                          */
/* C_C_LAST_RUN and C_C_LAST_LOAD must adhere to clause 2.1.6.               */
/*****************************************************************************/
#define C_C_LAST_RUN    88
#define C_C_LAST_LOAD   173
#define C_C_ID                          127
#define C_OL_I_ID       1723
#define A_C_LAST        255
#define A_C_ID                          1023
#define A_OL_I_ID       8191

/*****************************************************************************/
/* Transaction Type Identifiers                                              */
/*****************************************************************************/

#define NEWORD_SQL   1
#define PAYMENT_SQL  2
#define ORDSTAT_SQL  3
#define DELIVERY_SQL 4
#define STOCKLEV_SQL 5

#define SPGENERAL_PAD 3
#define SPGENERAL_ADJUST sizeof(int16_t)

struct in_neword_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  struct   in_items_struct {
    int32_t  s_OL_I_ID;
    int32_t  s_OL_SUPPLY_W_ID;
    int16_t  s_OL_QUANTITY;
    int16_t  pad1[3];
  } in_item[15];
  int64_t  s_O_ENTRY_D_time;            /* init by SUT */
  int32_t  s_C_ID;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
  int16_t  s_O_OL_CNT;                  /* init by SUT */
  int16_t  s_all_local;
  int16_t  duplicate_items;
};

struct out_neword_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  struct   items_struct {
    int32_t  s_I_PRICE;
    int32_t  s_OL_AMOUNT;
    int16_t  s_S_QUANTITY;
    int16_t  pad2;
    char     s_I_NAME[25];
    char     s_brand_generic;
  } item[15];
  int64_t  s_O_ENTRY_D_time;
  int32_t  s_W_TAX;
  int32_t  s_D_TAX;
  int32_t  s_C_DISCOUNT;
  int32_t  s_total_amount;
```

```
  int32_t  s_O_ID;
  int16_t  s_O_OL_CNT;
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_C_LAST[17];
  char     s_C_CREDIT[3];
};

struct in_payment_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int64_t  s_H_DATE_time;   /* init by SUT */
  int64_t  s_H_AMOUNT;
  int32_t  s_W_ID;
  int32_t  s_C_W_ID;
  int32_t  s_C_ID;
  int16_t  s_C_D_ID;
  int16_t  s_D_ID;
  char     s_C_LAST[17];
};

struct out_payment_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int64_t  s_H_DATE_time;
  int64_t  s_C_SINCE_time;
  int64_t  s_C_CREDIT_LIM;
  int64_t  s_C_BALANCE;
  int32_t  s_C_DISCOUNT;
  int32_t  s_C_ID;
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_W_STREET_1[21];
  char     s_W_STREET_2[21];
  char     s_W_CITY[21];
  char     s_W_STATE[3];
  char     s_W_ZIP[10];
  char     s_D_STREET_1[21];
  char     s_D_STREET_2[21];
  char     s_D_CITY[21];
  char     s_D_STATE[3];
  char     s_D_ZIP[10];
  char     s_C_FIRST[17];
  char     s_C_MIDDLE[3];
  char     s_C_LAST[17];
  char     s_C_STREET_1[21];
  char     s_C_STREET_2[21];
  char     s_C_CITY[21];
  char     s_C_STATE[3];
  char     s_C_ZIP[10];
  char     s_C_PHONE[17];
  char     s_C_CREDIT[3];
  char     s_C_DATA[201];
};

struct in_ordstat_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_C_ID;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
  int16_t  pad1[3];
  char     s_C_LAST[17];
};

struct out_ordstat_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int64_t  s_C_BALANCE;
  int64_t  s_O_ENTRY_D_time;
  int32_t  s_C_ID;
  int32_t  s_O_ID;
  int16_t  s_O_CARRIER_ID;
  int16_t  s_ol_cnt;
  int16_t  pad1[2];
  struct   oitems_struct {
    int64_t  s_OL_DELIVERY_D_time;
    int32_t  s_OL_AMOUNT;
    int32_t  s_OL_I_ID;
    int32_t  s_OL_SUPPLY_W_ID;
    int16_t  s_OL_QUANTITY;
    int16_t  pad2;
  } item[15];
  int16_t  s_transtatus;
  int16_t  deadlocks;
  char     s_C_FIRST[17];
  char     s_C_MIDDLE[3];
  char     s_C_LAST[17];
};

struct in_delivery_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int64_t  s_O_DELIVERY_D_time;          /* init by SUT */
  int32_t  s_W_ID;
  int16_t  s_O_CARRIER_ID;
};

struct out_delivery_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_O_ID[10];
  int16_t  s_transtatus;
  int16_t  deadlocks;
};

struct in_stocklev_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_threshold;
  int32_t  s_W_ID;
  int16_t  s_D_ID;
};

struct out_stocklev_struct {
  int16_t  len;
  int16_t  pad[SPGENERAL_PAD];
  int32_t  s_low_stock;
  int16_t  s_transtatus;
  int16_t  deadlocks;
};

/* ******************************************************************** */
/* Transaction Prototypes                                               */
/* ******************************************************************** */

#ifdef __cplusplus
extern "C" {
#endif

extern int neword_sql(struct in_neword_struct*, struct out_neword_struct*);
extern int payment_sql(struct in_payment_struct*, struct out_payment_struct*);
extern int ordstat_sql(struct in_ordstat_struct*, struct out_ordstat_struct*);
extern int delivery_sql(struct in_delivery_struct*, struct out_delivery_struct*);
extern int stocklev_sql(struct in_stocklev_struct*, struct out_stocklev_struct*);

#ifdef __cplusplus
}
#endif

/* ******************************************************************** */
/* Database Connect/Disconnect Prototypes                               */
/* ******************************************************************** */

#ifdef __cplusplus
extern "C" {
#endif

extern int connect_to_TM(char*);
extern int disconnect_from_TM(void);

#ifdef __cplusplus
}
#endif


#endif // __DB2TPCC_H
```

gendata.c

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * gendata.c - Generate data for TPC-C database
 */

#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <sqlutil.h>
/* UNIX named pipe support */
#include <sys/stat.h>
#include <errno.h>
#include <fcntl.h>

#include "platform.h"
#include "tpccrnd.h"
#include "lval.h"
```

```
/* PROTOTYPES. */
void gen_dist_tbl( void );
void gen_cust_tbl( void );
void gen_hist_tbl( void );
void gen_nu_ord_tbl( void );
void gen_ordr_tbl( void );
void gen_item_tbl( void );
void gen_stock_tbl( void );
void gen_ware_tbl( void );

int i, j;
double timestamp1, timestamp2, elapse;
int rc, rc1, rc2;

int using_range = 0;
int using_npipe = 0;
int using_rctload = 0;
sqlint32 ware_start=-1, ware_end=-1;

char fmtWare[]   = "%s|%s|%s|%s|%s|%s|%d|%lld|%d\n";
char fmtDist[]   = "%d|%d|%lld|%s|%s|%s|%s|%s|%s|%hd|%d\n";
char fmtItem[]   = "%s|%d|%s|%d|%d\n";
char fmtStock[]  = "%d|%d|%d|%d|%s|%s|%s|%s|%s|%s|%s|%s|%s|%s|%s|%d|%ld\n";
char fmtCust[]   = "%d|%s|%s|%s|%s|%lld|%lld|%s|%s|%d|%s|%s|%s|%s|%s|%s|%hd|%ld|%d|%lld|%lld|%d\n";
char fmtHist[]   = "%d|%hd|%ld|%hd|%ld|%lld|%d|%s\n";
char fmtOrdr[]   = "%d|%lld|%d|%d|%d|%d|%ld|%hd\n";
char fmtOLine[]  = "%lld|%d|%d|%ld|%hd|%s|%d|%hd|%ld|%hd\n";
char fmtNewOrd[] = "%d|%hd|%ld\n";
void InitFormatStrings(char delim);
void ScalingReport(void);

int outtype1 = 0;
int outtype2 = 0;
char *outname1 = NULL;
char *outname2 = NULL;

/*-------------------------------------------------------------------------*/
/*      main                                                               */
/*-------------------------------------------------------------------------*/
int main (int argc, char *argv[])
{
  int option = -1;
  char *delim = NULL;

  /* ******************************************************************* */
  /* Compute Warehouse Ranges                                            */
  /* ******************************************************************* */
  ware_start = 1;
  ware_end = WAREHOUSES;

  /* ******************************************************************* */
  /* Process Command Line Arguments                                      */
  /* ******************************************************************* */

  /* Valid Command Line Options
   * --------------------------
   * Table Option:            -t <table>        (-t 2 for warehouse)
   * Output Column Delimiter:  -d <char>         (-d ' ', -d '|', etc)
   * Output to File:          -f[n] <file>      (-f customer.dat)
   * Output to Pipe:          -p[n] <pipename>  (-p tpccpipe.000)
   * Warehouse Range:         -r <start> <end>  (-r 1 100)
   * Scaling Report:          -s
   *
   * The -f[n] and/or -p[n] options are required.
   * The -t, -d, -r and -s options are optional.
   *
   * If -d is omitted, the vertical bar (pipe) symbol ('|') will be used.
   * If -r is omitted, the range [1..WAREHOUSES] will be used.
   *
   * Due to the TPC-C spec requiring that orders and orderline be
   * generated at the same time, there is an extension to the -f and -p
   * options to specify one of the two output streams for each argument.
   *
   * -f1 orders.dat -f2 orderline.dat will output to two files
   * -f1 orders.dat -p2 tpccpipe.000 will output to a file and a pipe
   *
   * -f1/-p1 specifies the destination for the orders table
   * -f2/-p2 specifies the destination for the orderline table
   *
   */

  /* Read Arguments */
  for (i=1; i<argc; i++)
  {
    if (strcmp(argv[i], "-t") == 0) {
      option = atoi(argv[i+1]);
      i++;
    } else if (strcmp(argv[i], "-r") == 0) {
      ware_start = atoi(argv[i+1]);
      ware_end = atoi(argv[i+2]);
      i += 2;
    } else if (strcmp(argv[i], "-d") == 0) {
      delim = argv[i+1];
      i++;
    } else if ((strcmp(argv[i], "-f")  == 0) ||
               (strcmp(argv[i], "-f1") == 0)) {
      outtype1 = IOH_FILE;
      outname1 = argv[i+1];
      i++;
    } else if (strcmp(argv[i], "-f2") == 0) {
      outtype2 = IOH_FILE;
      outname2 = argv[i+1];
      i++;
    } else if ((strcmp(argv[i], "-p")  == 0) ||
               (strcmp(argv[i], "-p1") == 0)) {
      outtype1 = IOH_PIPE;
      outname1 = argv[i+1];
      i++;
    } else if (strcmp(argv[i], "-p2") == 0) {
      outtype2 = IOH_PIPE;
      outname2 = argv[i+1];
      i++;
    } else if (strcmp(argv[i], "-s") == 0) {
      ScalingReport();
      exit(0);
    } else {
      fprintf(stderr, "gendata: Don't understand argument: %s\n",argv[i]);
      exit(-1);
    }
  }

  /* ******************************************************************* */
  /* Validate Command Line Arguments                                     */
  /* ******************************************************************* */

  /* Validate Table Argument */
  if (option < 3 || option > 11 || option == 10)
  {
    fprintf(stderr,"gendata: Invalid table selected: %d\n",option);
    exit(-1);
  }

  /* Validate Delimiter Argument */
  if (delim == NULL) {
    // default delimiter is used for IMPORT & LOAD, no changes neccessary
    using_rctload = 0;
  } else if (strlen(delim) == 1 && !isalnum(delim[0]) &&
             delim[0] != '.' && delim[0] != '%')
  {
    // user-supplied delimiter used for rctload
    InitFormatStrings(delim[0]);
    using_rctload = 1;
  } else {
    fprintf(stderr,"gendata: Invalid delimiter specified: %s\n",delim);
    exit(-1);
  }

  /* Validate File/Pipe Arguments */
  if (option != 9 && outtype1 > 0 && outtype2 > 0)
  {
    fprintf(stderr,"gendata: Specifying two output file/pipes allowed only when
generating\norders/orderline.\n");
    exit(-1);
  }
  if (option == 9 && ((outtype1 == 0) || (outtype2 == 0)))
  {
    fprintf(stderr,"gendata: Must specify two output file/pipes when generating orders/orderline.\n");
    exit(-1);
  }
  if (outtype1 == 0 || outname1 == NULL || strcmp(outname1,"") == 0)
  {
    fprintf(stderr,"gendata: Invalid 1st output file/pipe specified.\n");
    exit(-1);
  }
  if (option == 9 && (outtype2 == 0 || outname2 == NULL || strcmp(outname2,"") == 0))
  {
    fprintf(stderr,"gendata: Invalid 2nd output file/pipe specified.\n");
    exit(-1);
  }
  /* Ensure O/OL flat files are opened in append mode.  This is required */
  /* because we generate O/OL concurrently.  See comments in genload.pl  */
  /* for further details on why this is neccessary.                      */
  if (option == 9)
  {
    if (outtype1 == IOH_FILE) outtype1 = IOH_FILE_APPEND;
    if (outtype2 == IOH_FILE) outtype2 = IOH_FILE_APPEND;
  }

  /* Validate Range Arguments */
  if (ware_start <= 0 || ware_start > WAREHOUSES) {
    fprintf(stderr,"gendata: Invalid range starting value: %d\n",ware_start);
     exit(-1);
  }
  if (ware_end <= 0 || ware_end > WAREHOUSES || ware_end < ware_start) {
    fprintf(stderr,"gendata: Invalid range ending value: %d\n",ware_end);
    exit(-1);
  }

  /* ******************************************************************* */
  /* Generate Data                                                       */
  /* ******************************************************************* */
  switch (option) {
   case 3: /* WAREHOUSE */
    gen_ware_tbl();
    break;
   case 4: /* DISTRICT  */
    gen_dist_tbl();
    break;
```

```
   case 5: /* ITEM  */
     gen_item_tbl();
     break;
   case 6: /* STOCK     */
     gen_stock_tbl();
     break;
   case 7: /* CUSTOMER  */
     gen_cust_tbl();
     break;
   case 8: /* HISTORY   */
     gen_hist_tbl();
     break;
   case 9: /* ORDERS + ORDER_LINE */
     gen_ordr_tbl();
     break;
   case 11: /* NEW_ORDER */
     gen_nu_ord_tbl();
     break;
   case 2:
   case 10:
   default:
     fprintf(stderr, "Error: invalid option = %d \n",(option));
     break;
  }
  return 0;
}
// MTE end

/*-----------------------------------------------------------------------*/
/* generate item table                                                   */
/*-----------------------------------------------------------------------*/

void gen_item_tbl( void )
{
  sqlint32  item_num ;
  sqlint32  item_im_id ;
  char  item_name[25] ;
  sqlint32  item_price ;
  char  item_data[51] ;

  IOH_NUM numBytes;
  ioHandle hnd;
  char Buffer[1024];

  initialize_random(13,42);
  timestamp1 = current_time();

  rc = GenericOpen(&hnd, outtype1, outname1);
  if (rc != 0) { goto item_done; }

  for(item_num = 1; item_num <= ITEMS; item_num++)
  {
    /* create image id field */
    item_im_id = rand_integer( 1, 10000 ) ;
    /* create name field */
    create_random_a_string( item_name, 14, 24);
    /* create price field */
    item_price = rand_integer( 100, 10000 ) ;
    /* create ORIGINAL field */
    create_a_string_with_original( item_data, 26, 50, 10) ;

    numBytes = sprintf(Buffer, fmtItem,
                       item_name,
                       item_price,
                       item_data,
                       item_im_id,
                       item_num);

    rc = GenericWrite(&hnd, Buffer, numBytes);
    if (rc != 0) { goto item_done; }

  } /* end for... */

  rc = GenericClose(&hnd);

item_done:

  timestamp2 = current_time();
  elapse = timestamp2 - timestamp1;
  if (rc == 0) {
    fprintf(stdout,"\nITEM table generated in %8.2f seconds.\n\n",elapse);
    fflush(stdout);
  } else {
    fprintf(stderr,"\nITEM table FAILED at (I %d) after %8.2f seconds.\n\n",item_num,elapse);
    fflush(stderr);
  }
}

/*-----------------------------------------------------------------------*/
/* generate stock table                                                  */
/*-----------------------------------------------------------------------*/
void gen_stock_tbl( void )
{
  sqlint32 ware_num;
  sqlint32 stock_num;
  sqlint32 stock_quant;
  sqlint32 s_ytd;
  sqlint32 s_order_cnt, s_remote_cnt;
  char  stock_dist_01[25] ;
  char  stock_dist_02[25] ;
  char  stock_dist_03[25] ;
  char  stock_dist_04[25] ;
  char  stock_dist_05[25] ;
  char  stock_dist_06[25] ;
  char  stock_dist_07[25] ;
  char  stock_dist_08[25] ;
  char  stock_dist_09[25] ;
  char  stock_dist_10[25] ;
  char  stock_data[51] ;

  IOH_NUM numBytes;
  ioHandle hnd;
  char Buffer[1024];

  initialize_random(7,11);
  timestamp1 = current_time();

  rc = GenericOpen(&hnd, outtype1, outname1);
  if (rc != 0) { goto stock_done; }

  for (stock_num = 1; stock_num <= STOCK_PER_WAREHOUSE; stock_num++)
  {
    if (stock_num%500 == 0)
    {
      fprintf(stdout, "STOCK for Item #%d\n",stock_num);
      fflush(stdout);
    }
    for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
    {
      stock_quant = rand_integer( 10, 100 ) ;
      create_random_a_string( stock_dist_01, 24, 24);
      create_random_a_string( stock_dist_02, 24, 24);
      create_random_a_string( stock_dist_03, 24, 24);
      create_random_a_string( stock_dist_04, 24, 24);
      create_random_a_string( stock_dist_05, 24, 24);
      create_random_a_string( stock_dist_06, 24, 24);
      create_random_a_string( stock_dist_07, 24, 24);
      create_random_a_string( stock_dist_08, 24, 24);
      create_random_a_string( stock_dist_09, 24, 24);
      create_random_a_string( stock_dist_10, 24, 24);

      /* create ORIGINAL field */
      create_a_string_with_original( stock_data, 26, 50, 10 );
      s_ytd = s_order_cnt = s_remote_cnt = 0;

      numBytes = sprintf(Buffer, fmtStock,
                         s_remote_cnt,
                         stock_quant,
                         s_order_cnt,
                         s_ytd,
                         stock_data,
                         stock_dist_01,
                         stock_dist_02,
                         stock_dist_03,
                         stock_dist_04,
                         stock_dist_05,
                         stock_dist_06,
                         stock_dist_07,
                         stock_dist_08,
                         stock_dist_09,
                         stock_dist_10,
                         stock_num,
                         ware_num);

      rc = GenericWrite(&hnd, Buffer, numBytes);
      if (rc != 0) { goto stock_done; }

    } /* end for... */
  } /* end for... */

  rc = GenericClose(&hnd);

stock_done:

  timestamp2 = current_time();
  elapse = timestamp2 - timestamp1;
  if (rc == 0) {
    fprintf(stdout,"\nSTOCK table generated in %8.2f seconds.\n\n",elapse);
    fflush(stdout);
  } else {
    fprintf(stderr,"\nSTOCK table FAILED at (S %d W %d) after %8.2f seconds.\n\n",stock_num,
ware_num, elapse);
    fflush(stderr);
  }
}

/*-----------------------------------------------------------------------*/
/* generate warehouse table                                              */
/*-----------------------------------------------------------------------*/
void gen_ware_tbl( void )
{
  sqlint32 ware_num;
  char ware_name[11] ;
  char ware_street_1[21] ;
  char ware_street_2[21] ;
  char ware_city[21] ;
  char ware_state[3]  ;
  char ware_zip[10]  ;
```

```
  sqlint32 ware_tax ;

  IOH_NUM numBytes;
  ioHandle hnd;
  char Buffer[1024];

  initialize_random(23,111);
  timestamp1 = current_time();

  rc = GenericOpen(&hnd, outtype1, outname1);
  if (rc != 0) { goto ware_done; }

  for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
  {
    /* ware_num is 0-based so we need to adjust when computing mod */
    if (((ware_num+1) % 500) == 0) {
      fprintf(stdout, "Warehouse #%d\n", ware_num);
      fflush(stdout);
    }

    create_random_a_string( ware_name,      6,10) ; /* create name */
    create_random_a_string( ware_street_1, 10,20) ; /* create street 1 */
    create_random_a_string( ware_street_2, 10,20) ; /* create street 2 */
    create_random_a_string( ware_city,     10,20) ; /* create city */
    create_random_a_string( ware_state,     2,2) ;  /* create state */
    create_random_n_string( ware_zip,       4,4) ;  /* create zip */
    strcat(ware_zip, "11111");

    ware_tax = rand_integer(0, 2000);

    numBytes = sprintf(Buffer, fmtWare,
                       ware_name,
                       ware_street_1,
                       ware_street_2,
                       ware_city,
                       ware_state,
                       ware_zip,
                       ware_tax,
                       30000000LL,
                       ware_num);

    rc = GenericWrite(&hnd, Buffer, numBytes);
    if (rc != 0) { goto ware_done; }

  } /* end for */

  rc = GenericClose(&hnd);

ware_done:

  timestamp2 = current_time();
  elapse = timestamp2 - timestamp1;
  if (rc == 0) {
    fprintf(stdout,"\nWAREHOUSE table generated in %8.2f seconds.\n\n",elapse);
    fflush(stdout);
  } else {
    fprintf(stderr,"\nWAREHOUSE table FAILED at (W %d) after %8.2f seconds.\n\n",ware_num, elapse);
    fflush(stderr);
  }
}

/*---------------------------------------------------------------------*/
/* generate dist table                                                 */
/*---------------------------------------------------------------------*/
void gen_dist_tbl( void )
{
  sqlint32 ware_num;
  sqlint32 dist_num;
  char dist_name[11];
  char dist_street_1[21];
  char dist_street_2[21];
  char dist_city[21];
  char dist_state[3];
  char dist_zip[10];
  sqlint32 dist_tax;
  sqlint32 next_o_id;

  IOH_NUM numBytes;
  ioHandle hnd;
  char Buffer[1024];

  next_o_id = CUSTOMERS_PER_DISTRICT + 1;
  initialize_random(44,73);
  timestamp1 = current_time();

  rc = GenericOpen(&hnd, outtype1, outname1);
  if (rc != 0) { goto dist_done; }

  for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
  {
    fprintf(stdout, "DISTRICT for Warehouse #%d\n", ware_num);
    fflush(stdout);
    for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE; dist_num++)
    {
      create_random_a_string( dist_name,      6,10) ; /* create name */
      create_random_a_string( dist_street_1, 10,20) ; /* create street 1 */
      create_random_a_string( dist_street_2, 10,20) ; /* create street 2 */
      create_random_a_string( dist_city,     10,20) ; /* create city */
      create_random_a_string( dist_state,     2,2) ;  /* create state */
      create_random_n_string( dist_zip,       4,4) ;  /* create zip */
      strcat(dist_zip, "11111");
      dist_tax = rand_integer(0, 2000);

      numBytes = sprintf(Buffer, fmtDist,
                         next_o_id,
                         dist_tax,
                         3000000LL,
                         dist_name,
                         dist_street_1,
                         dist_street_2,
                         dist_city,
                         dist_state,
                         dist_zip,
                         dist_num,
                         ware_num);

      rc = GenericWrite(&hnd, Buffer, numBytes);
      if (rc != 0) { goto dist_done; }

    } /* end for... */
  } /* end for... */

  rc = GenericClose(&hnd);

dist_done:

  timestamp2 = current_time();
  elapse = timestamp2 - timestamp1;
  if (rc == 0) {
    fprintf(stdout,"\nDISTRICT table generated in %8.2f seconds.\n\n",elapse);
    fflush(stdout);
  } else {
    fprintf(stderr,"\nDISTRICT table FAILED at (W %d D %d) after %8.2f
seconds.\n\n",ware_num,dist_num,elapse);
    fflush(stderr);
  }
}

/*---------------------------------------------------------------------*/
/* generate customer table                                             */
/*---------------------------------------------------------------------*/
void gen_cust_tbl( void )
{
  sqlint32 ware_num;
  sqlint32 dist_num;
  sqlint32 cust_num;
  char cust_last[17];
  char cust_middle[3];
  char cust_first[17];
  char cust_street_1[21];
  char cust_street_2[21];
  char cust_city[21];
  char cust_state[3];
  char cust_zip[10];
  char cust_phone[17];
  char cust_credit[3];
  char cust_data[501];
  sqlint32 cust_discount;
  sqlint64 currtmstmp;

  IOH_NUM numBytes;
  ioHandle hnd;
  char Buffer[1024];
  int len, pos;

  initialize_random(10,64);
  timestamp1 = current_time();

  rc = GenericOpen(&hnd, outtype1, outname1);
  if (rc != 0) { goto cust_done; }

  strcpy(cust_middle, "OE");
  currtmstmp = time(NULL);

  for (cust_num = 1; cust_num <= CUSTOMERS_PER_DISTRICT; cust_num++)
  {
    fprintf(stdout, "CUSTOMER #%d:\n", cust_num);
    fflush(stdout);

    for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
    {
      for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE; dist_num++)
      {
        if (cust_num <= 1000)                   /* create last name */
          create_random_last_name( cust_last, cust_num);
        else                                    /* create last name */
          create_random_last_name( cust_last, 0);
        create_random_a_string( cust_first,     8,16) ; /* create first name */
        create_random_a_string( cust_street_1, 10,20) ; /* create street 1 */
        create_random_a_string( cust_street_2, 10,20) ; /* create street 2 */
        create_random_a_string( cust_city,     10,20) ; /* create city */
        create_random_a_string( cust_state,     2,2) ;  /* create state */
        create_random_n_string( cust_zip,       4,4) ;  /* create zip */
        strcat(cust_zip, "11111");

        /* create phone number */
        create_random_n_string( cust_phone, 16,16) ;
        if ( rand_integer( 1, 100 ) <= 10 )
```

```
          strcpy( cust_credit, "BC" ) ;
        else
          strcpy( cust_credit, "GC" ) ;

        /* create discount rate */
        cust_discount = rand_integer(0, 5000);

        /* create customer data */
        create_random_a_string(cust_data, 300, 500);

        /* pad customer data (only for non-rctload) */
        if (using_rctload == 0) {
          for (pos=strlen(cust_data); pos<500; pos++)
            cust_data[pos] = ' ';
          cust_data[500] = '\0';
        }

        numBytes = sprintf(Buffer, fmtCust,
                           cust_num,
                           cust_state,
                           cust_zip,
                           cust_phone,
                           currtmstmp,
                           5000000LL,
                           cust_middle,
                           cust_credit,
                           cust_discount,
                           cust_data,
                           cust_last,
                           cust_first,
                           cust_street_1,
                           cust_street_2,
                           cust_city,
                           dist_num,
                           ware_num,
                           0,
                           -1000LL,
                           1000LL,
                           1);

        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto cust_done; }

      } /* end for district... */
    } /* end for warehouse... */
  } /* end for customer... */

  rc = GenericClose(&hnd);

cust_done:

  timestamp2 = current_time();
  elapse = timestamp2 - timestamp1;
  if (rc == 0) {
    fprintf(stdout,"\nCUSTOMER table generated in %8.2f seconds.\n\n",elapse);
    fflush(stdout);
  } else {
    fprintf(stderr,"\nCUSTOMER table FAILED at (W %d D %d C %d) after %8.2f
seconds.\n\n",ware_num, dist_num, cust_num, elapse);
    fflush(stderr);
  }
}

/*-------------------------------------------------------------------------*/
/* generate hist table                                                     */
/*-------------------------------------------------------------------------*/
void gen_hist_tbl( void )
{
  sqlint32 ware_num;
  sqlint32 dist_num;
  sqlint32 cust_num;
  char hist_data[25] ;
  sqlint64 currtmstmp;

  IOH_NUM numBytes;
  ioHandle hnd;
  char Buffer[1024];

  initialize_random(15,63);
  timestamp1 = current_time();

  rc = GenericOpen(&hnd, outtype1, outname1);
  if (rc != 0) { goto hist_done; }

  currtmstmp = time(NULL);

  for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
  {
    fprintf(stdout, "HISTORY for Warehouse #%d:\n", ware_num);
    fflush(stdout);
    for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE; dist_num++)
    {
      for (cust_num = 1; cust_num <= CUSTOMERS_PER_DISTRICT; cust_num++)
      {
        /* create history data */
        create_random_a_string( hist_data, 12,24) ;

        numBytes = sprintf(Buffer, fmtHist,
                           cust_num,
                           dist_num,
                           ware_num,
                           dist_num,
                           ware_num,
                           currtmstmp,
                           1000,
                           hist_data);

        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto hist_done; }

      } /* end for customer... */
    } /* end for district... */
  } /* end for warehouse... */

  rc = GenericClose(&hnd);

hist_done:

  timestamp2 = current_time();
  elapse = timestamp2 - timestamp1;
  if (rc == 0) {
    fprintf(stdout,"\nHISTORY table generated in %8.2f seconds.\n\n",elapse);
    fflush(stdout);
  } else {
    fprintf(stderr,"\nHISTORY table FAILED at (W %d D %d C %d) after %8.2f seconds.\n\n",ware_num,
dist_num, cust_num, elapse);
    fflush(stderr);
  }
}

/*-------------------------------------------------------------------------*/
/* generate nu_ord table                                                   */
/*-------------------------------------------------------------------------*/
void gen_nu_ord_tbl( void )
{
  sqlint32 ware_num;
  sqlint32 dist_num;
  sqlint32 nu_ord_id ;
  int nu_ord_hi ;

  IOH_NUM numBytes;
  ioHandle hnd;
  char Buffer[1024];

  /* compute maximum and minimum
     order numbers for this
     district */
  nu_ord_hi = CUSTOMERS_PER_DISTRICT - NU_ORDERS_PER_DISTRICT + 1;
  if (nu_ord_hi < 0) {
    nu_ord_hi = CUSTOMERS_PER_DISTRICT - (CUSTOMERS_PER_DISTRICT / 3) + 1;
    fprintf(stderr,"\n**** WARNING **** NU_ORDERS_PER_DISTRICT is >
CUSTOMERS_PER_DISTRICT\n");
    fprintf(stderr,"          Check the values in file lval.h\n");
    fprintf(stderr,"          Loading New-Order with 1/3 of CUSTOMERS_PER_DISTRICT\n");
  }
  initialize_random(99,37);
  timestamp1 = current_time();

  rc = GenericOpen(&hnd, outtype1, outname1);
  if (rc != 0) { goto neword_done; }

  for (nu_ord_id = nu_ord_hi;
       nu_ord_id <= CUSTOMERS_PER_DISTRICT;
       nu_ord_id++)
  {
    fprintf(stdout, "NEW_ORDER for Customer #%d:\n", nu_ord_id);
    fflush(stdout);
    for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
    {
      for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE; dist_num++)
      {
        numBytes = sprintf(Buffer, fmtNewOrd,
                           nu_ord_id,
                           dist_num,
                           ware_num);

        rc = GenericWrite(&hnd, Buffer, numBytes);
        if (rc != 0) { goto neword_done; }

      } /* end for... */
    } /* end for... */
  } /* end for... */

  rc = GenericClose(&hnd);

neword_done:

  timestamp2 = current_time();
  elapse = timestamp2 - timestamp1;
  if (rc == 0) {
    fprintf(stdout,"\nNEW_ORDER table generated in %8.2f seconds.\n\n",elapse);
    fflush(stdout);
  } else {
    fprintf(stderr,"\nNEW_ORDER table FAILED at (W %d D %d O %d) after %8.2f
seconds.\n\n",ware_num, dist_num, nu_ord_id, elapse);
    fflush(stderr);
  }
}
```

```
/*------------------------------------------------------------------------*/
/* generate order and order_line tables                                   */
/*------------------------------------------------------------------------*/
void gen_ordr_tbl( void )
{
  sqlint32 ware_num ;
  sqlint32 dist_num ;
  sqlint32 cust_num ;
  sqlint32 ord_num ;
  sqlint32 ordr_carrier_id;
  sqlint32 ordr_ol_cnt;
  sqlint32 oline_ol_num;
  sqlint32 oline_item_num;

  sqlint32 oline_amount;
  char oline_dist_info[25];
  sqlint64 nulltmstmp = 0;
  sqlint64 currtmstmp;

  IOH_NUM numBytes;
  ioHandle hnd1, hnd2;
  char Buffer[1024];

  oline_dist_info[24] = '\0';
  initialize_random(42,13);
  timestamp1 = current_time();

  rc1 = GenericOpen(&hnd1, outtype1, outname1);
  if (rc1 != 0) { goto ool_done; }
  rc2 = GenericOpen(&hnd2, outtype2, outname2);
  if (rc2 != 0) { goto ool_done; }

  currtmstmp = time(NULL);

  for (ware_num = ware_start; ware_num <= ware_end; ware_num++)
  {
    fprintf(stdout, "ORDERS & ORDER_LINE for Warehouse #%d\n", ware_num);
    fflush(stdout);

    for (dist_num = 1; dist_num <= DISTRICTS_PER_WAREHOUSE; dist_num++)
    {
      fprintf(stdout, "District #%d\t", dist_num);
      fflush(stdout);

      seed_1_3000();

      for (ord_num = 1; ord_num <= CUSTOMERS_PER_DISTRICT; ord_num++)
      {
        if (ord_num < 2101)
          ordr_carrier_id = rand_integer( 1, 10 ) ;
        else
          ordr_carrier_id = 0;

        cust_num = random_1_3000();
        ordr_ol_cnt = rand_integer(MIN_OL_PER_ORDER,MAX_OL_PER_ORDER);

        numBytes = sprintf(Buffer, fmtOrdr,
                           cust_num,
                           currtmstmp,
                           ordr_carrier_id,
                           ordr_ol_cnt,
                           1,
                           ord_num,
                           ware_num,
                           dist_num);

        rc1 = GenericWrite(&hnd1, Buffer, numBytes);
        if (rc1 != 0) { goto ool_done; }

        for ( oline_ol_num = 1; oline_ol_num <= ordr_ol_cnt; oline_ol_num++ )
        {
          oline_item_num = rand_integer(1, ITEMS) ;
          create_random_a_string( oline_dist_info, 24, 24) ;

          numBytes = sprintf(Buffer, fmtOLine,
                             ((ord_num < 2101) ? currtmstmp : nulltmstmp),
                             ((ord_num < 2101) ? 0 : rand_integer(1,999999)),
                             oline_item_num,
                             ware_num,
                             5,
                             oline_dist_info,
                             ord_num,
                             dist_num,
                             ware_num,
                             oline_ol_num);

          rc2 = GenericWrite(&hnd2, Buffer, numBytes);
          if (rc2 != 0) { goto ool_done; }

        } /* for order_line */
      } /* for order */
    } /* for dist */
  } /* for ware */

  rc1 = GenericClose(&hnd2);
  rc2 = GenericClose(&hnd1);

ool_done:
  timestamp2 = current_time();
  elapse = timestamp2 - timestamp1;
  if (rc1 == 0 && rc2 == 0) {
    fprintf(stdout,"\nORDERS & ORDER_LINE table(s) generated in %8.2f seconds.\n\n",elapse);
    fflush(stdout);
  } else {
    fprintf(stderr,"\nORDERS & ORDER_LINE table(s) FAILED at (W %d D %d O %d OL %d) after
%8.2f seconds.\n\n",ware_num, dist_num, ord_num, oline_ol_num, elapse);
    fflush(stderr);
  }
}

// This routine will initalize the printf format strings and replace the
// delimiter with the one provided.  The pipe symbol is the default.
void InitFormatStrings(char delim)
{
  char *p;

  // Check if Using Default Delimiter
  if (delim == '|') return;

  // Replace Delimiters
  while (p = strchr(fmtWare,'|'))   { *p = delim; }
  while (p = strchr(fmtDist,'|'))   { *p = delim; }
  while (p = strchr(fmtItem,'|'))   { *p = delim; }
  while (p = strchr(fmtStock,'|'))  { *p = delim; }
  while (p = strchr(fmtCust,'|'))   { *p = delim; }
  while (p = strchr(fmtHist,'|'))   { *p = delim; }
  while (p = strchr(fmtOrdr,'|'))   { *p = delim; }
  while (p = strchr(fmtOLine,'|'))  { *p = delim; }
  while (p = strchr(fmtNewOrd,'|')) { *p = delim; }
}

void ScalingReport(void)
{
  /* Print Scaling Values */
  fprintf(stdout,"Scaling Values in Use\n");
  fprintf(stdout,"-----------------------------\n");
  fprintf(stdout,"Warehouses:           %d\n", WAREHOUSES);
  fprintf(stdout,"Districts/Warehouse:   %d\n", DISTRICTS_PER_WAREHOUSE);
  fprintf(stdout,"Customers/District:    %d\n", CUSTOMERS_PER_DISTRICT);
  fprintf(stdout,"Items:                 %d\n", ITEMS);
  fprintf(stdout,"Stock/Warehouse:       %d\n", STOCK_PER_WAREHOUSE);
  fprintf(stdout,"Min Order Lines/Order: %d\n", MIN_OL_PER_ORDER);
  fprintf(stdout,"Max Order Lines/Order: %d\n", MAX_OL_PER_ORDER);
  fprintf(stdout,"New Orders/District:   %d\n", NU_ORDERS_PER_DISTRICT);
  fprintf(stdout,"-----------------------------\n");

}
```

lval.h

```
/* lval.h - generated automatically at 20030505.0618 */

#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 53200
#define DISTRICTS_PER_WAREHOUSE 10
#define CUSTOMERS_PER_DISTRICT 3000
#define ITEMS 100000
#define STOCK_PER_WAREHOUSE 100000
#define MIN_OL_PER_ORDER 5
#define MAX_OL_PER_ORDER 15
#define NU_ORDERS_PER_DISTRICT 900

#endif // __LVAL_H
```

platform.h

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * platform.h - Platform Isolation Layer
 */

#ifndef __PLATFORM_H
#define __PLATFORM_H

/* ***************************************************************************/
```

```
/* Generic Macros                                          */
/* ************************************************************************/
#define GEN_ERRCODE     errno

/* ************************************************************************/
/* Windows I/O Macros                                      */
/* ************************************************************************/

/* ************************************************************************/
/* UNIX I/O Macros                                         */
/* ************************************************************************/
#include <fcntl.h>

#define IOH_INIT(hnd, type, name)                           \
  hnd->fd = -1;                                             \
  hnd->type = type;                                         \
  hnd->name = name;

#define IOH_CREATE(hnd)                                     \
  if (hnd->type == IOH_PIPE) {                              \
    rc = mkfifo(hnd->name, 0666);                           \
  } else {                                                  \
    rc = 0;                                                 \
  }

#define IOH_OPEN(hnd)                                       \
  if (hnd->type == IOH_FILE_APPEND) {                       \
    hnd->fd = open(hnd->name, O_WRONLY | O_CREAT | O_APPEND, 0666);     \
  } else {                                                  \
    hnd->fd = open(hnd->name, O_WRONLY | O_CREAT | O_TRUNC, 0666);      \
  }                                                         \
  if (hnd->fd == -1) {                                      \
    rc = -1;                                                \
  } else {                                                  \
    rc = 0;                                                 \
  }

#define IOH_WRITE(hnd, buff, num, num2)                     \
  rc = write(hnd->fd, buff, num);                           \
  if (rc >= 0) {                                            \
    num2 = rc;                                              \
    rc = 0;                                                 \
  }

#define IOH_FLUSH(hnd)    rc = 0;
#define IOH_CLOSE(hnd)    rc = close(hnd->fd);
#define IOH_DELETE(hnd)  if (hnd->type == IOH_PIPE) { rc = unlink(hnd->name); }

typedef unsigned int IOH_NUM;
typedef int IOH_HND;

/*************************************************************************/
/* UNIX Semaphore Macros                                   */
/*************************************************************************/
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/sem.h>

union semun {
  int val;
  struct semid_ds *buf;
  unsigned short int *array;
} semUnion;

struct sembuf semBuf;

#define SEM_HANDLE int

#define SEM_INIT(hnd, x, name)                              \
  if ( (hnd = semget(IPC_PRIVATE, 1 , IPC_CREAT | IPC_EXCL | S_IRUSR | S_IWUSR | S_IRGRP |
S_IWGRP | S_IROTH | S_IWOTH)) == -1)                        \
    API_ERROR(__LINE__, "semget", (rc=GEN_ERRCODE));        \
  semUnion.val = x;                                         \
  if ( semctl(hnd, 0, SETVAL, semUnion) < 0 )               \
    API_ERROR(__LINE__, "semctl SETVAL", (rc=GEN_ERRCODE));

#define SEM_WAIT(hnd)                                       \
  semBuf.sem_num = 0;                                       \
  semBuf.sem_op = -1;                                       \
  semBuf.sem_flg = SEM_UNDO;                                \
  if ( semop(hnd, &semBuf, 1) < 0 )                         \
    API_ERROR(__LINE__, "semop wait", (rc=GEN_ERRCODE));

#define SEM_FREE(hnd)                                       \
  semBuf.sem_num = 0;                                       \
  semBuf.sem_op = 1;                                        \
  semBuf.sem_flg = SEM_UNDO;                                \
  if ( semop(hnd, &semBuf, 1) < 0 )                         \
    API_ERROR(__LINE__, "semop free", (rc=GEN_ERRCODE));

#define SEM_DESTROY(hnd)                                    \
  if ( semctl(hnd, 0, IPC_RMID, 0) )                        \
    API_ERROR(__LINE__, "semctl IPC_RMID", (rc=GEN_ERRCODE));


/* ************************************************************************/
/* Common I/O Macros and Definitions                       */
/* ************************************************************************/
#define IOH_FILE 1
#define IOH_PIPE 2
#define IOH_FILE_APPEND 3

#define IOH_ERRMSG(hnd, msg)                                \
  if (rc != 0) {                                            \
    fprintf(stderr, "Error %d %s fd %d (%d, %s)\n", GEN_ERRCODE, msg,     \
                hnd->fd, hnd->type, hnd->name);             \
    return rc;                                              \
  }

struct _ioh {
  IOH_HND fd;
  int type;
  char *name;
};

typedef struct _ioh ioHandle;

/* ************************************************************************/
/* Generic I/O Routine Prototypes                          */
/* ************************************************************************/
int GenericOpen(ioHandle *hnd, int type, char *name);
int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes);
int GenericClose(ioHandle *hnd);

/* ************************************************************************/
/* Generic I/O Routines                                    */
/* ************************************************************************/
int GenericOpen(ioHandle *hnd, int type, char *name)
{
  int rc = 0;

  IOH_INIT(hnd, type, name)

  IOH_CREATE(hnd)
  IOH_ERRMSG(hnd, "creating")

  IOH_OPEN(hnd)
  IOH_ERRMSG(hnd, "opening")

  return rc;
}

int GenericWrite(ioHandle *hnd, char *Buffer, unsigned int numBytes)
{
  int rc = 0;
  int numBytesWritten = -1;

  IOH_WRITE(hnd, Buffer, numBytes, numBytesWritten)
  IOH_ERRMSG(hnd, "writing")
  if (numBytes != numBytesWritten) {
    fprintf(stderr, "Truncated data writing to fd %d (%d, %s)\n", hnd->fd, hnd->type, hnd->name);
    rc = -1;
  }
  return rc;
}

int GenericClose(ioHandle *hnd)
{
  int rc = 0;

  IOH_FLUSH(hnd)
  IOH_ERRMSG(hnd, "flushing")

  IOH_CLOSE(hnd)
  IOH_ERRMSG(hnd, "closing")

  IOH_DELETE(hnd)
  IOH_ERRMSG(hnd, "deleting")

  return rc;
}


#endif // __PLATFORM_H
```

tpccrnd.c

```
/*************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*************************************************************************/

/*
```

```
 * tpccrnd.c - Random generation functions for TPC-C
 */

#include "db2tpcc.h"
#include "lval.h"

static char tbl_cust[CUSTOMERS_PER_DISTRICT];

static char alnum[] =
  "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";

static char *last_name_parts[] =
{
  "BAR",
  "OUGHT",
  "ABLE",
  "PRI",
  "PRES",
  "ESE",
  "ANTI",
  "CALLY",
  "ATION",
  "EING"
};

/*
*******************************************************************************
*   seed_1_3000
*
*
*******************************************************************************
*/

void seed_1_3000( void )
{
 int i;

 for (i = 0; i < CUSTOMERS_PER_DISTRICT; i++) {
   tbl_cust[i] = 0;
 }
}

/*
*******************************************************************************
*   random_1_3000
*
*
*******************************************************************************
*/

int random_1_3000( void )
{
 static int i;
 static int x;

 x = rand_integer(0, CUSTOMERS_PER_DISTRICT - 1);

 for (i = 0; i < CUSTOMERS_PER_DISTRICT; i++)
 {
   if (tbl_cust[x] == 0)
   {
     tbl_cust[x] = 1;
     return(x+1);
   } else {
     x++;
   }
   if (x == CUSTOMERS_PER_DISTRICT)
     x=0;
 }

 printf("\nfatal error in random_1_3000 \n");
 abort();
}


/*
*******************************************************************************
*   initialize_random
*
*
*******************************************************************************
*/

void initialize_random(int x, int y)
{
 srand(x);
 srandom(y);
}


/*
*******************************************************************************
*   create_random_a_string
*
*   create a random alphanumeric string, of random length between lo and
*   hi and place them in designated buffer. Routine returns the actual
*   length.
*
*   parameters
*   ----------
*   lo end of acceptable length range
*   hi end of acceptable length range
*
*   output
*   ------
*   actual length
*   random alphanumeric string
*
*******************************************************************************
*/

int create_random_a_string( char *out_buffer, int length_lo, int length_hi )
{
 int i, actual_length ;

 actual_length = rand_integer( length_lo, length_hi ) ;

 for (i = 0; i < actual_length; i++ )
 {
  out_buffer[i] = alnum[rand_integer( 0, 61 )] ;
 }
 out_buffer[actual_length] = '\0' ;

 return (actual_length);
}


/*
*******************************************************************************
*   create_random_n_string
*
*   create a random numeric string, of random length between lo and
*   hi and place them in designated buffer. Routine returns the actual
*   length.
*
*   parameters
*   ----------
*   lo end of acceptable length range
*   hi end of acceptable length range
*
*   output
*   ------
*   actual length
*   random numeric string
*
*******************************************************************************
*/

int create_random_n_string( char *out_buffer, int length_lo, int length_hi )
{
 int i, actual_length ;

 actual_length = rand_integer( length_lo, length_hi ) ;

 for (i = 0; i < actual_length; i++ )
 {
  out_buffer[i] = (char)rand_integer( 48,57 ) ;
 }
 out_buffer[actual_length] = '\0' ;

 return (actual_length);
}

/*
*******************************************************************************
*   NUrand_val
*
*   create a non-uniform random numeric value of type integer, of random
*   value between lo and hi. Number is NOT placed in BUFFER, and IS
*   simply RETURNED.
*
*   Routine RETURNS the VALUE.
*
*   parameters
*   ----------
*   lo end of acceptable value range
*   hi end of acceptable value range
*
*   output
*   ------
*   random integer value  RETURNED
*
*******************************************************************************
*/

int NUrand_val ( int A, int x, int y, int C )
{
 return((((rand_integer(0,A)|rand_integer(x,y))+C)%(y-x+1))+x);
}


/*
*******************************************************************************
*   rand_integer
*
*   create a uniform random numeric value of type integer, of random
*   value between lo and hi. Number is NOT placed in BUFFER, and IS
*   simply RETURNED.
```

```
*
*   Routine RETURNS the VALUE.
*
*   parameters
*   ----------
*   lo end of acceptable value range
*   hi end of acceptable value range
*
*   output
*   ------
*   random integer value  RETURNED
*
*****************************************************************************
*/

int rand_integer ( int val_lo, int val_hi )
{
 return((random()%(val_hi-val_lo+1))+val_lo);
}

/*
*****************************************************************************
*   create_a_string_with_original
*
*   create a random alphanumeric string, of random length between lo and
*   hi and place them in designated buffer. Routine returns the actual
*   length.
*
*   the word "ORIGINAL" is placed at a random location in the buffer at
*   random, for a given percent of the records.
*
*   percent_to_set must be an integer value from 0 to 100.
*   if 0, no records will be set. If 100, all records will be set.
*
*   CANNOT USE ON STRINGS OF LENGTH LESS THAN 8 ! LOWER LIMIT MUST BE > 8 !
*
*   parameters
*   ----------
*   lo end of acceptable length range
*   hi end of acceptable length range
*   percentage of records to set to ORIGINAL
*
*   output
*   ------
*   actual length
*   random alphanumeric string with the word "ORIGINAL" is placed at a
*      random location
*
*****************************************************************************
*/

int create_a_string_with_original( char *out_buffer, int length_lo,
                                   int length_hi, int percent_to_set )
{
 int actual_length, start_pos ;

 actual_length = create_random_a_string( out_buffer, length_lo, length_hi ) ;

 if ( rand_integer( 1, 100 ) <= percent_to_set )
 {
  start_pos = rand_integer( 0, actual_length-8 ) ;
  strncpy(out_buffer+start_pos,"ORIGINAL",8);
 }

 return (actual_length);
}


/*****************************************************************************
 *
 * create_random_last_name
 *
 * parameters:
 *    out_buffer - target buffer for the generated last name
 *
 * description:
 *    create_random_last_name 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. the three strings are
 *    concatenated to generate the name
 *
 *****************************************************************************/

int create_random_last_name(char *out_buffer, int cust_num)
{
 int random_num;

 if (cust_num == 0)
  random_num = NUrand_val( A_C_LAST, 0, 999, C_C_LAST_LOAD );
 else
  random_num = cust_num - 1;

 strcpy(out_buffer, last_name_parts[random_num / 100]);
 random_num %= 100;
 strcat(out_buffer, last_name_parts[random_num / 10]);
 random_num %= 10;
 strcat(out_buffer, last_name_parts[random_num]);

 return(strlen(out_buffer));
}
```

tpccrnd.h

```
/*****************************************************************************
** Licensed Materials - Property of IBM
**
** Governed under the terms of the International
** License Agreement for Non-Warranted Sample Code.
**
** (C) COPYRIGHT International Business Machines Corp. 1996 - 2002
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*****************************************************************************/

/*
 * tpccrnd.h - Random generation functions for TPC-C
 */

#ifndef __TPCCRND_H
#define __TPCCRND_H

void initialize_random(int x, int y);
int rand_integer( int val_lo, int val_hi ) ;
int NUrand_val( int A, int val_lo, int val_hi, int C ) ;

void seed_1_3000( void );
int random_1_3000( void );

int create_random_a_string( char *out_buffer,
                            int length_lo,
                            int length_hi );
int create_random_n_string( char *out_buffer,
                            int length_lo,
                            int length_hi );
int create_a_string_with_original( char *out_buffer,
                                   int length_lo,
                                   int length_hi,
                                   int percent_to_set ) ;
int create_random_last_name(char *out_buffer, int cust_num);


#endif // __TPCCRND_H
```

Appendix D: RTE Scripts

D.1 RTE Parameters

```
/* 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 0
#define MASTER_NUM6 0
#define MASTER_NUM7 0
#define MASTER_NUM8 0
#define MASTER_NUM9 0

#if MASTER_NUM1
MASTER "master1"
#elif MASTER_NUM2
MASTER "master2"
#elif MASTER_NUM3
MASTER "master3"
#elif MASTER_NUM4
MASTER "master4"
#elif MASTER_NUM5
MASTER "master5"
#elif MASTER_NUM6
MASTER "master6"
#elif MASTER_NUM7
MASTER "master7"
#elif MASTER_NUM8
MASTER "master8"
#elif MASTER_NUM9
MASTER "master9"
#endif
/*---- SUT -----------------------------------------------*/
SUT="toraus"
/*--------------------------------------------------------*/
LASTC=88
MEASUREMENT="1"
WAREHOUSES=53200
/*----- SLAVES ------------------------------------------*/

#if MASTER_NUM1
SLAVES driver1a, driver1b, driver1c, driver1d, driver1e, driver1f, driver2a, driver2b, driver2c, driver2d,
driver2e, driver2f, driver3a, driver3b, driver3c, driver3d, driver3e, driver3f, driver4a, driver4b, driver4c,
driver4d, driver4e, driver4f, driver5a, driver5b, driver5c, driver5d, driver5e, driver5f, driver6a, driver6b,
driver6c, driver6d, driver6e, driver6f
#elif MASTER_NUM2
SLAVES driver7a, driver7b, driver7c, driver7d, driver7e, driver7f, driver8a, driver8b, driver8c, driver8d,
driver8e, driver8f, driver9a, driver9b, driver9c, driver9d, driver9e, driver9f, driver10a, driver10b, driver10c,
driver10d, driver10e, driver10f, driver11a, driver11b, driver11c, driver11d, driver11e, driver11f, driver12a,
driver12b, driver12c, driver12d, driver12e, driver12f
#elif MASTER_NUM3
SLAVES driver13a, driver13b, driver13c, driver13d, driver13e, driver13f, driver14a, driver14b, driver14c,
driver14d, driver14e, driver14f, driver15a, driver15b, driver15c, driver15d, driver15e, driver15f, driver16a,
driver16b, driver16c, driver16d, driver16e, driver16f, driver17a, driver17b, driver17c, driver17d, driver17e,
driver17f, driver18a, driver18b, driver18c, driver18d, driver18e, driver18f
#elif MASTER_NUM4
SLAVES driver19a, driver19b, driver19c, driver19d, driver19e, driver19f, driver20a, driver20b, driver20c,
driver20d, driver20e, driver20f, driver21a, driver21b, driver21c, driver21d, driver21e, driver21f, driver22a,
driver22b, driver22c, driver22d, driver22e, driver22f, driver23a, driver23b, driver23c, driver23d, driver23e,
driver23f, driver24a, driver24b, driver24c, driver24d, driver24e, driver24f
#elif MASTER_NUM5
SLAVES driver25a, driver25b, driver25c, driver25d, driver25e, driver25f, driver26a, driver26b, driver26c,
driver26d, driver26e, driver26f, driver27a, driver27b, driver27c, driver27d, driver27e, driver27f, driver28a,
driver28b, driver28c, driver28d, driver28e, driver28f, driver29a, driver29b, driver29c, driver29d, driver29e,
driver29f, driver30a, driver30b, driver30c, driver30d, driver30e, driver30f
#elif MASTER_NUM6
SLAVES driver31a, driver31b, driver31c, driver31d, driver31e, driver31f, driver32a, driver32b, driver32c,
driver32d, driver32e, driver32f, driver33a, driver33b, driver33c, driver33d, driver33e, driver33f, driver34a,
driver34b, driver34c, driver34d, driver34e, driver34f, driver35a, driver35b, driver35c, driver35d, driver35e,
driver35f, driver36a, driver36b, driver36c, driver36d, driver36e, driver36f
#elif MASTER_NUM7
SLAVES driver37a, driver37b, driver37c, driver37d, driver37e, driver37f, driver38a, driver38b, driver38c,
driver38d, driver38e, driver38f, driver39a, driver39b, driver39c, driver39d, driver39e, driver39f, driver40a,
driver40b, driver40c, driver40d, driver40e, driver40f, driver41a, driver41b, driver41c, driver41d, driver41e,
driver41f, driver42a, driver42b, driver42c, driver42d, driver42e, driver42f
#elif MASTER_NUM8
SLAVES driver43a, driver43b, driver43c, driver43d, driver43e, driver43f, driver44a, driver44b, driver44c,
driver44d, driver44e, driver44f, driver45a, driver45b, driver45c, driver45d, driver45e, driver45f, driver46a,
driver46b, driver46c, driver46d, driver46e, driver46f, driver47a, driver47b, driver47c, driver47d, driver47e,
driver47f, driver48a, driver48b, driver48c, driver48d, driver48e, driver48f
#elif MASTER_NUM9
SLAVES driver49a, driver49b, driver49c, driver49d, driver49e, driver49f, driver50a, driver50b, driver50c,
driver50d, driver50e, driver50f, driver51a, driver51b, driver51c, driver51d, driver51e, driver51f, driver52a,
driver52b, driver52c, driver52d, driver52e, driver52f, driver53a, driver53b, driver53c, driver53d, driver53e,
driver53f, driver54a, driver54b, driver54c, driver54d, driver54e, driver54f
#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"
#endif

/*---- more client ctuff ------------------------------*/
#if MASTER_NUM1
CLIENT client1a tpcc tpcc
CLIENT client1b tpcc tpcc
CLIENT client1c tpcc tpcc
CLIENT client1d tpcc tpcc
CLIENT client1e tpcc tpcc
CLIENT client1f tpcc tpcc
CLIENT client1g tpcc tpcc
CLIENT client1h tpcc tpcc
CLIENT client1i tpcc tpcc
CLIENT client1j tpcc tpcc
CLIENT client1k tpcc tpcc
CLIENT client1l tpcc tpcc
CLIENT client1m tpcc tpcc
CLIENT client1n tpcc tpcc
CLIENT client1o tpcc tpcc
CLIENT client1p tpcc tpcc
CLIENT client1q tpcc tpcc
CLIENT client1r tpcc tpcc
CLIENT client1s tpcc tpcc
CLIENT client1t tpcc tpcc
CLIENT client1u tpcc tpcc
CLIENT client1v tpcc tpcc
CLIENT client1w tpcc tpcc
CLIENT client1x tpcc tpcc
CLIENT client1y tpcc tpcc
CLIENT client1z tpcc tpcc
CLIENT client1aa tpcc tpcc
CLIENT client1ab tpcc tpcc

CLIENT client2a tpcc tpcc
CLIENT client2b tpcc tpcc
CLIENT client2c tpcc tpcc
CLIENT client2d tpcc tpcc
CLIENT client2e tpcc tpcc
CLIENT client2f tpcc tpcc
CLIENT client2g tpcc tpcc
CLIENT client2h tpcc tpcc
CLIENT client2i tpcc tpcc
CLIENT client2j tpcc tpcc
CLIENT client2k tpcc tpcc
CLIENT client2l tpcc tpcc
CLIENT client2m tpcc tpcc
CLIENT client2n tpcc tpcc
CLIENT client2o tpcc tpcc
CLIENT client2p tpcc tpcc
CLIENT client2q tpcc tpcc
CLIENT client2r tpcc tpcc
CLIENT client2s tpcc tpcc
CLIENT client2t tpcc tpcc
CLIENT client2u tpcc tpcc
CLIENT client2v tpcc tpcc
CLIENT client2w tpcc tpcc
CLIENT client2x tpcc tpcc
CLIENT client2y tpcc tpcc
CLIENT client2z tpcc tpcc
CLIENT client2aa tpcc tpcc
CLIENT client2ab tpcc tpcc

CLIENT client3a tpcc tpcc
CLIENT client3b tpcc tpcc
CLIENT client3c tpcc tpcc
CLIENT client3d tpcc tpcc
CLIENT client3e tpcc tpcc
CLIENT client3f tpcc tpcc
CLIENT client3g tpcc tpcc
```

```
CLIENT client3h tpcc tpcc
CLIENT client3i tpcc tpcc
CLIENT client3j tpcc tpcc
CLIENT client3k tpcc tpcc
CLIENT client3l tpcc tpcc
CLIENT client3m tpcc tpcc
CLIENT client3n tpcc tpcc
CLIENT client3o tpcc tpcc
CLIENT client3p tpcc tpcc
CLIENT client3q tpcc tpcc
CLIENT client3r tpcc tpcc
CLIENT client3s tpcc tpcc
CLIENT client3t tpcc tpcc
CLIENT client3u tpcc tpcc
CLIENT client3v tpcc tpcc
CLIENT client3w tpcc tpcc
CLIENT client3x tpcc tpcc
CLIENT client3y tpcc tpcc
CLIENT client3z tpcc tpcc
CLIENT client3aa tpcc tpcc
CLIENT client3ab tpcc tpcc

#elif MASTER_NUM2
CLIENT client4a tpcc tpcc
CLIENT client4b tpcc tpcc
CLIENT client4c tpcc tpcc
CLIENT client4d tpcc tpcc
CLIENT client4e tpcc tpcc
CLIENT client4f tpcc tpcc
CLIENT client4g tpcc tpcc
CLIENT client4h tpcc tpcc
CLIENT client4i tpcc tpcc
CLIENT client4j tpcc tpcc
CLIENT client4k tpcc tpcc
CLIENT client4l tpcc tpcc
CLIENT client4m tpcc tpcc
CLIENT client4n tpcc tpcc
CLIENT client4o tpcc tpcc
CLIENT client4p tpcc tpcc
CLIENT client4q tpcc tpcc
CLIENT client4r tpcc tpcc
CLIENT client4s tpcc tpcc
CLIENT client4t tpcc tpcc
CLIENT client4u tpcc tpcc
CLIENT client4v tpcc tpcc
CLIENT client4w tpcc tpcc
CLIENT client4x tpcc tpcc
CLIENT client4y tpcc tpcc
CLIENT client4z tpcc tpcc
CLIENT client4aa tpcc tpcc
CLIENT client4ab tpcc tpcc

CLIENT client5a tpcc tpcc
CLIENT client5b tpcc tpcc
CLIENT client5c tpcc tpcc
CLIENT client5d tpcc tpcc
CLIENT client5e tpcc tpcc
CLIENT client5f tpcc tpcc
CLIENT client5g tpcc tpcc
CLIENT client5h tpcc tpcc
CLIENT client5i tpcc tpcc
CLIENT client5j tpcc tpcc
CLIENT client5k tpcc tpcc
CLIENT client5l tpcc tpcc
CLIENT client5m tpcc tpcc
CLIENT client5n tpcc tpcc
CLIENT client5o tpcc tpcc
CLIENT client5p tpcc tpcc
CLIENT client5q tpcc tpcc
CLIENT client5r tpcc tpcc
CLIENT client5s tpcc tpcc
CLIENT client5t tpcc tpcc
CLIENT client5u tpcc tpcc
CLIENT client5v tpcc tpcc
CLIENT client5w tpcc tpcc
CLIENT client5x tpcc tpcc
CLIENT client5y tpcc tpcc
CLIENT client5z tpcc tpcc
CLIENT client5aa tpcc tpcc
CLIENT client5ab tpcc tpcc

CLIENT client6a tpcc tpcc
CLIENT client6b tpcc tpcc
CLIENT client6c tpcc tpcc
CLIENT client6d tpcc tpcc
CLIENT client6e tpcc tpcc
CLIENT client6f tpcc tpcc
CLIENT client6g tpcc tpcc
CLIENT client6h tpcc tpcc
CLIENT client6i tpcc tpcc
CLIENT client6j tpcc tpcc
CLIENT client6k tpcc tpcc
CLIENT client6l tpcc tpcc
CLIENT client6m tpcc tpcc
CLIENT client6n tpcc tpcc
CLIENT client6o tpcc tpcc
CLIENT client6p tpcc tpcc
CLIENT client6q tpcc tpcc
CLIENT client6r tpcc tpcc
CLIENT client6s tpcc tpcc
CLIENT client6t tpcc tpcc
CLIENT client6u tpcc tpcc
CLIENT client6v tpcc tpcc
CLIENT client6w tpcc tpcc
CLIENT client6x tpcc tpcc
CLIENT client6y tpcc tpcc
CLIENT client6z tpcc tpcc
CLIENT client6aa tpcc tpcc
CLIENT client6ab tpcc tpcc

#elif MASTER_NUM3
CLIENT client7a tpcc tpcc
CLIENT client7b tpcc tpcc
CLIENT client7c tpcc tpcc
CLIENT client7d tpcc tpcc
CLIENT client7e tpcc tpcc
CLIENT client7f tpcc tpcc
CLIENT client7g tpcc tpcc
CLIENT client7h tpcc tpcc
CLIENT client7i tpcc tpcc
CLIENT client7j tpcc tpcc
CLIENT client7k tpcc tpcc
CLIENT client7l tpcc tpcc
CLIENT client7m tpcc tpcc
CLIENT client7n tpcc tpcc
CLIENT client7o tpcc tpcc
CLIENT client7p tpcc tpcc
CLIENT client7q tpcc tpcc
CLIENT client7r tpcc tpcc
CLIENT client7s tpcc tpcc
CLIENT client7t tpcc tpcc
CLIENT client7u tpcc tpcc
CLIENT client7v tpcc tpcc
CLIENT client7w tpcc tpcc
CLIENT client7x tpcc tpcc
CLIENT client7y tpcc tpcc
CLIENT client7z tpcc tpcc
CLIENT client7aa tpcc tpcc
CLIENT client7ab tpcc tpcc

CLIENT client8a tpcc tpcc
CLIENT client8b tpcc tpcc
CLIENT client8c tpcc tpcc
CLIENT client8d tpcc tpcc
CLIENT client8e tpcc tpcc
CLIENT client8f tpcc tpcc
CLIENT client8g tpcc tpcc
CLIENT client8h tpcc tpcc
CLIENT client8i tpcc tpcc
CLIENT client8j tpcc tpcc
CLIENT client8k tpcc tpcc
CLIENT client8l tpcc tpcc
CLIENT client8m tpcc tpcc
CLIENT client8n tpcc tpcc
CLIENT client8o tpcc tpcc
CLIENT client8p tpcc tpcc
CLIENT client8q tpcc tpcc
CLIENT client8r tpcc tpcc
CLIENT client8s tpcc tpcc
CLIENT client8t tpcc tpcc
CLIENT client8u tpcc tpcc
CLIENT client8v tpcc tpcc
CLIENT client8w tpcc tpcc
CLIENT client8x tpcc tpcc
CLIENT client8y tpcc tpcc
CLIENT client8z tpcc tpcc
CLIENT client8aa tpcc tpcc
CLIENT client8ab tpcc tpcc

CLIENT client9a tpcc tpcc
CLIENT client9b tpcc tpcc
CLIENT client9c tpcc tpcc
CLIENT client9d tpcc tpcc
CLIENT client9e tpcc tpcc
CLIENT client9f tpcc tpcc
CLIENT client9g tpcc tpcc
CLIENT client9h tpcc tpcc
CLIENT client9i tpcc tpcc
CLIENT client9j tpcc tpcc
CLIENT client9k tpcc tpcc
CLIENT client9l tpcc tpcc
CLIENT client9m tpcc tpcc
CLIENT client9n tpcc tpcc
CLIENT client9o tpcc tpcc
CLIENT client9p tpcc tpcc
CLIENT client9q tpcc tpcc
CLIENT client9r tpcc tpcc
CLIENT client9s tpcc tpcc
CLIENT client9t tpcc tpcc
CLIENT client9u tpcc tpcc
CLIENT client9v tpcc tpcc
CLIENT client9w tpcc tpcc
CLIENT client9x tpcc tpcc
CLIENT client9y tpcc tpcc
CLIENT client9z tpcc tpcc
CLIENT client9aa tpcc tpcc
CLIENT client9ab tpcc tpcc
```

```
#elif MASTER_NUM4
CLIENT client10a tpcc tpcc
CLIENT client10b tpcc tpcc
CLIENT client10c tpcc tpcc
CLIENT client10d tpcc tpcc
CLIENT client10e tpcc tpcc
CLIENT client10f tpcc tpcc
CLIENT client10g tpcc tpcc
CLIENT client10h tpcc tpcc
CLIENT client10i tpcc tpcc
CLIENT client10j tpcc tpcc
CLIENT client10k tpcc tpcc
CLIENT client10l tpcc tpcc
CLIENT client10m tpcc tpcc
CLIENT client10n tpcc tpcc
CLIENT client10o tpcc tpcc
CLIENT client10p tpcc tpcc
CLIENT client10q tpcc tpcc
CLIENT client10r tpcc tpcc
CLIENT client10s tpcc tpcc
CLIENT client10t tpcc tpcc
CLIENT client10u tpcc tpcc
CLIENT client10v tpcc tpcc
CLIENT client10w tpcc tpcc
CLIENT client10x tpcc tpcc
CLIENT client10y tpcc tpcc
CLIENT client10z tpcc tpcc
CLIENT client10aa tpcc tpcc
CLIENT client10ab tpcc tpcc

CLIENT client11a tpcc tpcc
CLIENT client11b tpcc tpcc
CLIENT client11c tpcc tpcc
CLIENT client11d tpcc tpcc
CLIENT client11e tpcc tpcc
CLIENT client11f tpcc tpcc
CLIENT client11g tpcc tpcc
CLIENT client11h tpcc tpcc
CLIENT client11i tpcc tpcc
CLIENT client11j tpcc tpcc
CLIENT client11k tpcc tpcc
CLIENT client11l tpcc tpcc
CLIENT client11m tpcc tpcc
CLIENT client11n tpcc tpcc
CLIENT client11o tpcc tpcc
CLIENT client11p tpcc tpcc
CLIENT client11q tpcc tpcc
CLIENT client11r tpcc tpcc
CLIENT client11s tpcc tpcc
CLIENT client11t tpcc tpcc
CLIENT client11u tpcc tpcc
CLIENT client11v tpcc tpcc
CLIENT client11w tpcc tpcc
CLIENT client11x tpcc tpcc
CLIENT client11y tpcc tpcc
CLIENT client11z tpcc tpcc
CLIENT client11aa tpcc tpcc
CLIENT client11ab tpcc tpcc

CLIENT client12a tpcc tpcc
CLIENT client12b tpcc tpcc
CLIENT client12c tpcc tpcc
CLIENT client12d tpcc tpcc
CLIENT client12e tpcc tpcc
CLIENT client12f tpcc tpcc
CLIENT client12g tpcc tpcc
CLIENT client12h tpcc tpcc
CLIENT client12i tpcc tpcc
CLIENT client12j tpcc tpcc
CLIENT client12k tpcc tpcc
CLIENT client12l tpcc tpcc
CLIENT client12m tpcc tpcc
CLIENT client12n tpcc tpcc
CLIENT client12o tpcc tpcc
CLIENT client12p tpcc tpcc
CLIENT client12q tpcc tpcc
CLIENT client12r tpcc tpcc
CLIENT client12s tpcc tpcc
CLIENT client12t tpcc tpcc
CLIENT client12u tpcc tpcc
CLIENT client12v tpcc tpcc
CLIENT client12w tpcc tpcc
CLIENT client12x tpcc tpcc
CLIENT client12y tpcc tpcc
CLIENT client12z tpcc tpcc
CLIENT client12aa tpcc tpcc
CLIENT client12ab tpcc tpcc

#elif MASTER_NUM5
CLIENT client13a tpcc tpcc
CLIENT client13b tpcc tpcc
CLIENT client13c tpcc tpcc
CLIENT client13d tpcc tpcc
CLIENT client13e tpcc tpcc
CLIENT client13f tpcc tpcc
CLIENT client13g tpcc tpcc
CLIENT client13h tpcc tpcc
CLIENT client13i tpcc tpcc
CLIENT client13j tpcc tpcc
CLIENT client13k tpcc tpcc
CLIENT client13l tpcc tpcc
CLIENT client13m tpcc tpcc
CLIENT client13n tpcc tpcc
CLIENT client13o tpcc tpcc
CLIENT client13p tpcc tpcc
CLIENT client13q tpcc tpcc
CLIENT client13r tpcc tpcc
CLIENT client13s tpcc tpcc
CLIENT client13t tpcc tpcc
CLIENT client13u tpcc tpcc
CLIENT client13v tpcc tpcc
CLIENT client13w tpcc tpcc
CLIENT client13x tpcc tpcc
CLIENT client13y tpcc tpcc
CLIENT client13z tpcc tpcc
CLIENT client13aa tpcc tpcc
CLIENT client13ab tpcc tpcc

CLIENT client14a tpcc tpcc
CLIENT client14b tpcc tpcc
CLIENT client14c tpcc tpcc
CLIENT client14d tpcc tpcc
CLIENT client14e tpcc tpcc
CLIENT client14f tpcc tpcc
CLIENT client14g tpcc tpcc
CLIENT client14h tpcc tpcc
CLIENT client14i tpcc tpcc
CLIENT client14j tpcc tpcc
CLIENT client14k tpcc tpcc
CLIENT client14l tpcc tpcc
CLIENT client14m tpcc tpcc
CLIENT client14n tpcc tpcc
CLIENT client14o tpcc tpcc
CLIENT client14p tpcc tpcc
CLIENT client14q tpcc tpcc
CLIENT client14r tpcc tpcc
CLIENT client14s tpcc tpcc
CLIENT client14t tpcc tpcc
CLIENT client14u tpcc tpcc
CLIENT client14v tpcc tpcc
CLIENT client14w tpcc tpcc
CLIENT client14x tpcc tpcc
CLIENT client14y tpcc tpcc
CLIENT client14z tpcc tpcc
CLIENT client14aa tpcc tpcc
CLIENT client14ab tpcc tpcc

CLIENT client15a tpcc tpcc
CLIENT client15b tpcc tpcc
CLIENT client15c tpcc tpcc
CLIENT client15d tpcc tpcc
CLIENT client15e tpcc tpcc
CLIENT client15f tpcc tpcc
CLIENT client15g tpcc tpcc
CLIENT client15h tpcc tpcc
CLIENT client15i tpcc tpcc
CLIENT client15j tpcc tpcc
CLIENT client15k tpcc tpcc
CLIENT client15l tpcc tpcc
CLIENT client15m tpcc tpcc
CLIENT client15n tpcc tpcc
CLIENT client15o tpcc tpcc
CLIENT client15p tpcc tpcc
CLIENT client15q tpcc tpcc
CLIENT client15r tpcc tpcc
CLIENT client15s tpcc tpcc
CLIENT client15t tpcc tpcc
CLIENT client15u tpcc tpcc
CLIENT client15v tpcc tpcc
CLIENT client15w tpcc tpcc
CLIENT client15x tpcc tpcc
CLIENT client15y tpcc tpcc
CLIENT client15z tpcc tpcc
CLIENT client15aa tpcc tpcc
CLIENT client15ab tpcc tpcc

#elif MASTER_NUM6
CLIENT client16a tpcc tpcc
CLIENT client16b tpcc tpcc
CLIENT client16c tpcc tpcc
CLIENT client16d tpcc tpcc
CLIENT client16e tpcc tpcc
CLIENT client16f tpcc tpcc
CLIENT client16g tpcc tpcc
CLIENT client16h tpcc tpcc
CLIENT client16i tpcc tpcc
CLIENT client16j tpcc tpcc
CLIENT client16k tpcc tpcc
CLIENT client16l tpcc tpcc
CLIENT client16m tpcc tpcc
CLIENT client16n tpcc tpcc
CLIENT client16o tpcc tpcc
CLIENT client16p tpcc tpcc
CLIENT client16q tpcc tpcc
CLIENT client16r tpcc tpcc
CLIENT client16s tpcc tpcc
CLIENT client16t tpcc tpcc
CLIENT client16u tpcc tpcc
```

```
CLIENT client16v tpcc tpcc
CLIENT client16w tpcc tpcc
CLIENT client16x tpcc tpcc
CLIENT client16y tpcc tpcc
CLIENT client16z tpcc tpcc
CLIENT client16aa tpcc tpcc
CLIENT client16ab tpcc tpcc

CLIENT client17a tpcc tpcc
CLIENT client17b tpcc tpcc
CLIENT client17c tpcc tpcc
CLIENT client17d tpcc tpcc
CLIENT client17e tpcc tpcc
CLIENT client17f tpcc tpcc
CLIENT client17g tpcc tpcc
CLIENT client17h tpcc tpcc
CLIENT client17i tpcc tpcc
CLIENT client17j tpcc tpcc
CLIENT client17k tpcc tpcc
CLIENT client17l tpcc tpcc
CLIENT client17m tpcc tpcc
CLIENT client17n tpcc tpcc
CLIENT client17o tpcc tpcc
CLIENT client17p tpcc tpcc
CLIENT client17q tpcc tpcc
CLIENT client17r tpcc tpcc
CLIENT client17s tpcc tpcc
CLIENT client17t tpcc tpcc
CLIENT client17u tpcc tpcc
CLIENT client17v tpcc tpcc
CLIENT client17w tpcc tpcc
CLIENT client17x tpcc tpcc
CLIENT client17y tpcc tpcc
CLIENT client17z tpcc tpcc
CLIENT client17aa tpcc tpcc
CLIENT client17ab tpcc tpcc

CLIENT client18a tpcc tpcc
CLIENT client18b tpcc tpcc
CLIENT client18c tpcc tpcc
CLIENT client18d tpcc tpcc
CLIENT client18e tpcc tpcc
CLIENT client18f tpcc tpcc
CLIENT client18g tpcc tpcc
CLIENT client18h tpcc tpcc
CLIENT client18i tpcc tpcc
CLIENT client18j tpcc tpcc
CLIENT client18k tpcc tpcc
CLIENT client18l tpcc tpcc
CLIENT client18m tpcc tpcc
CLIENT client18n tpcc tpcc
CLIENT client18o tpcc tpcc
CLIENT client18p tpcc tpcc
CLIENT client18q tpcc tpcc
CLIENT client18r tpcc tpcc
CLIENT client18s tpcc tpcc
CLIENT client18t tpcc tpcc
CLIENT client18u tpcc tpcc
CLIENT client18v tpcc tpcc
CLIENT client18w tpcc tpcc
CLIENT client18x tpcc tpcc
CLIENT client18y tpcc tpcc
CLIENT client18z tpcc tpcc
CLIENT client18aa tpcc tpcc
CLIENT client18ab tpcc tpcc

#elif MASTER_NUM7
CLIENT client19a tpcc tpcc
CLIENT client19b tpcc tpcc
CLIENT client19c tpcc tpcc
CLIENT client19d tpcc tpcc
CLIENT client19e tpcc tpcc
CLIENT client19f tpcc tpcc
CLIENT client19g tpcc tpcc
CLIENT client19h tpcc tpcc
CLIENT client19i tpcc tpcc
CLIENT client19j tpcc tpcc
CLIENT client19k tpcc tpcc
CLIENT client19l tpcc tpcc
CLIENT client19m tpcc tpcc
CLIENT client19n tpcc tpcc
CLIENT client19o tpcc tpcc
CLIENT client19p tpcc tpcc
CLIENT client19q tpcc tpcc
CLIENT client19r tpcc tpcc
CLIENT client19s tpcc tpcc
CLIENT client19t tpcc tpcc
CLIENT client19u tpcc tpcc
CLIENT client19v tpcc tpcc
CLIENT client19w tpcc tpcc
CLIENT client19x tpcc tpcc
CLIENT client19y tpcc tpcc
CLIENT client19z tpcc tpcc
CLIENT client19aa tpcc tpcc
CLIENT client19ab tpcc tpcc

CLIENT client20a tpcc tpcc
CLIENT client20b tpcc tpcc
CLIENT client20c tpcc tpcc
CLIENT client20d tpcc tpcc
CLIENT client20e tpcc tpcc
CLIENT client20f tpcc tpcc
CLIENT client20g tpcc tpcc
CLIENT client20h tpcc tpcc
CLIENT client20i tpcc tpcc
CLIENT client20j tpcc tpcc
CLIENT client20k tpcc tpcc
CLIENT client20l tpcc tpcc
CLIENT client20m tpcc tpcc
CLIENT client20n tpcc tpcc
CLIENT client20o tpcc tpcc
CLIENT client20p tpcc tpcc
CLIENT client20q tpcc tpcc
CLIENT client20r tpcc tpcc
CLIENT client20s tpcc tpcc
CLIENT client20t tpcc tpcc
CLIENT client20u tpcc tpcc
CLIENT client20v tpcc tpcc
CLIENT client20w tpcc tpcc
CLIENT client20x tpcc tpcc
CLIENT client20y tpcc tpcc
CLIENT client20z tpcc tpcc
CLIENT client20aa tpcc tpcc
CLIENT client20ab tpcc tpcc

CLIENT client21a tpcc tpcc
CLIENT client21b tpcc tpcc
CLIENT client21c tpcc tpcc
CLIENT client21d tpcc tpcc
CLIENT client21e tpcc tpcc
CLIENT client21f tpcc tpcc
CLIENT client21g tpcc tpcc
CLIENT client21h tpcc tpcc
CLIENT client21i tpcc tpcc
CLIENT client21j tpcc tpcc
CLIENT client21k tpcc tpcc
CLIENT client21l tpcc tpcc
CLIENT client21m tpcc tpcc
CLIENT client21n tpcc tpcc
CLIENT client21o tpcc tpcc
CLIENT client21p tpcc tpcc
CLIENT client21q tpcc tpcc
CLIENT client21r tpcc tpcc
CLIENT client21s tpcc tpcc
CLIENT client21t tpcc tpcc
CLIENT client21u tpcc tpcc
CLIENT client21v tpcc tpcc
CLIENT client21w tpcc tpcc
CLIENT client21x tpcc tpcc
CLIENT client21y tpcc tpcc
CLIENT client21z tpcc tpcc
CLIENT client21aa tpcc tpcc
CLIENT client21ab tpcc tpcc

#elif MASTER_NUM8
CLIENT client22a tpcc tpcc
CLIENT client22b tpcc tpcc
CLIENT client22c tpcc tpcc
CLIENT client22d tpcc tpcc
CLIENT client22e tpcc tpcc
CLIENT client22f tpcc tpcc
CLIENT client22g tpcc tpcc
CLIENT client22h tpcc tpcc
CLIENT client22i tpcc tpcc
CLIENT client22j tpcc tpcc
CLIENT client22k tpcc tpcc
CLIENT client22l tpcc tpcc
CLIENT client22m tpcc tpcc
CLIENT client22n tpcc tpcc
CLIENT client22o tpcc tpcc
CLIENT client22p tpcc tpcc
CLIENT client22q tpcc tpcc
CLIENT client22r tpcc tpcc
CLIENT client22s tpcc tpcc
CLIENT client22t tpcc tpcc
CLIENT client22u tpcc tpcc
CLIENT client22v tpcc tpcc
CLIENT client22w tpcc tpcc
CLIENT client22x tpcc tpcc
CLIENT client22y tpcc tpcc
CLIENT client22z tpcc tpcc
CLIENT client22aa tpcc tpcc
CLIENT client22ab tpcc tpcc

CLIENT client23a tpcc tpcc
CLIENT client23b tpcc tpcc
CLIENT client23c tpcc tpcc
CLIENT client23d tpcc tpcc
CLIENT client23e tpcc tpcc
CLIENT client23f tpcc tpcc
CLIENT client23g tpcc tpcc
CLIENT client23h tpcc tpcc
CLIENT client23i tpcc tpcc
CLIENT client23j tpcc tpcc
CLIENT client23k tpcc tpcc
CLIENT client23l tpcc tpcc
CLIENT client23m tpcc tpcc
CLIENT client23n tpcc tpcc
```

```
CLIENT client23o tpcc tpcc
CLIENT client23p tpcc tpcc
CLIENT client23q tpcc tpcc
CLIENT client23r tpcc tpcc
CLIENT client23s tpcc tpcc
CLIENT client23t tpcc tpcc
CLIENT client23u tpcc tpcc
CLIENT client23v tpcc tpcc
CLIENT client23w tpcc tpcc
CLIENT client23x tpcc tpcc
CLIENT client23y tpcc tpcc
CLIENT client23z tpcc tpcc
CLIENT client23aa tpcc tpcc
CLIENT client23ab tpcc tpcc

CLIENT client24a tpcc tpcc
CLIENT client24b tpcc tpcc
CLIENT client24c tpcc tpcc
CLIENT client24d tpcc tpcc
CLIENT client24e tpcc tpcc
CLIENT client24f tpcc tpcc
CLIENT client24g tpcc tpcc
CLIENT client24h tpcc tpcc
CLIENT client24i tpcc tpcc
CLIENT client24j tpcc tpcc
CLIENT client24k tpcc tpcc
CLIENT client24l tpcc tpcc
CLIENT client24m tpcc tpcc
CLIENT client24n tpcc tpcc
CLIENT client24o tpcc tpcc
CLIENT client24p tpcc tpcc
CLIENT client24q tpcc tpcc
CLIENT client24r tpcc tpcc
CLIENT client24s tpcc tpcc
CLIENT client24t tpcc tpcc
CLIENT client24u tpcc tpcc
CLIENT client24v tpcc tpcc
CLIENT client24w tpcc tpcc
CLIENT client24x tpcc tpcc
CLIENT client24y tpcc tpcc
CLIENT client24z tpcc tpcc
CLIENT client24aa tpcc tpcc
CLIENT client24ab tpcc tpcc

#elif MASTER_NUM9
CLIENT client25a tpcc tpcc
CLIENT client25b tpcc tpcc
CLIENT client25c tpcc tpcc
CLIENT client25d tpcc tpcc
CLIENT client25e tpcc tpcc
CLIENT client25f tpcc tpcc
CLIENT client25g tpcc tpcc
CLIENT client25h tpcc tpcc
CLIENT client25i tpcc tpcc
CLIENT client25j tpcc tpcc
CLIENT client25k tpcc tpcc
CLIENT client25l tpcc tpcc
CLIENT client25m tpcc tpcc
CLIENT client25n tpcc tpcc
CLIENT client25o tpcc tpcc
CLIENT client25p tpcc tpcc
CLIENT client25q tpcc tpcc
CLIENT client25r tpcc tpcc
CLIENT client25s tpcc tpcc
CLIENT client25t tpcc tpcc
CLIENT client25u tpcc tpcc
CLIENT client25v tpcc tpcc
CLIENT client25w tpcc tpcc
CLIENT client25x tpcc tpcc
CLIENT client25y tpcc tpcc
CLIENT client25z tpcc tpcc
CLIENT client25aa tpcc tpcc
CLIENT client25ab tpcc tpcc

CLIENT client26a tpcc tpcc
CLIENT client26b tpcc tpcc
CLIENT client26c tpcc tpcc
CLIENT client26d tpcc tpcc
CLIENT client26e tpcc tpcc
CLIENT client26f tpcc tpcc
CLIENT client26g tpcc tpcc
CLIENT client26h tpcc tpcc
CLIENT client26i tpcc tpcc
CLIENT client26j tpcc tpcc
CLIENT client26k tpcc tpcc
CLIENT client26l tpcc tpcc
CLIENT client26m tpcc tpcc
CLIENT client26n tpcc tpcc
CLIENT client26o tpcc tpcc
CLIENT client26p tpcc tpcc
CLIENT client26q tpcc tpcc
CLIENT client26r tpcc tpcc
CLIENT client26s tpcc tpcc
CLIENT client26t tpcc tpcc
CLIENT client26u tpcc tpcc
CLIENT client26v tpcc tpcc
CLIENT client26w tpcc tpcc
CLIENT client26x tpcc tpcc
CLIENT client26y tpcc tpcc
CLIENT client26z tpcc tpcc
CLIENT client26aa tpcc tpcc
CLIENT client26ab tpcc tpcc

CLIENT client27a tpcc tpcc
CLIENT client27b tpcc tpcc
CLIENT client27c tpcc tpcc
CLIENT client27d tpcc tpcc
CLIENT client27e tpcc tpcc
CLIENT client27f tpcc tpcc
CLIENT client27g tpcc tpcc
CLIENT client27h tpcc tpcc
CLIENT client27i tpcc tpcc
CLIENT client27j tpcc tpcc
CLIENT client27k tpcc tpcc
CLIENT client27l tpcc tpcc
CLIENT client27m tpcc tpcc
CLIENT client27n tpcc tpcc
CLIENT client27o tpcc tpcc
CLIENT client27p tpcc tpcc
CLIENT client27q tpcc tpcc
CLIENT client27r tpcc tpcc
CLIENT client27s tpcc tpcc
CLIENT client27t tpcc tpcc
CLIENT client27u tpcc tpcc
CLIENT client27v tpcc tpcc
CLIENT client27w tpcc tpcc
CLIENT client27x tpcc tpcc
CLIENT client27y tpcc tpcc
CLIENT client27z tpcc tpcc
CLIENT client27aa tpcc tpcc
CLIENT client27ab tpcc tpcc

#endif
/*--------------------------------------------------------------*/
TELNET telnet 23
SOCKET socket 199703
/* --- Sockets ------------------------------------------------*/
#if MASTER_NUM1
SOCKET_NETWORK socket1   6500 driver1a
SOCKET_NETWORK socket2   6501 driver1b
SOCKET_NETWORK socket3   6502 driver1c
SOCKET_NETWORK socket4   6503 driver1d
SOCKET_NETWORK socket5   6504 driver1e
SOCKET_NETWORK socket6   6505 driver1f
SOCKET_NETWORK socket7   6506 driver1a
SOCKET_NETWORK socket8   6507 driver1b
SOCKET_NETWORK socket9   6508 driver1c
SOCKET_NETWORK socket10   6509 driver1d
SOCKET_NETWORK socket11   6510 driver1e
SOCKET_NETWORK socket12   6511 driver1f
SOCKET_NETWORK socket13   6512 driver1a
SOCKET_NETWORK socket14   6513 driver1b
SOCKET_NETWORK socket15   6514 driver1c
SOCKET_NETWORK socket16   6515 driver1d
SOCKET_NETWORK socket17   6516 driver1e
SOCKET_NETWORK socket18   6517 driver1f
SOCKET_NETWORK socket19   6518 driver1a
SOCKET_NETWORK socket20   6519 driver1b
SOCKET_NETWORK socket21   6520 driver1c
SOCKET_NETWORK socket22   6521 driver1d
SOCKET_NETWORK socket23   6522 driver1e
SOCKET_NETWORK socket24   6523 driver1f
SOCKET_NETWORK socket25   6524 driver1a
SOCKET_NETWORK socket26   6525 driver1b
SOCKET_NETWORK socket27   6526 driver1c
SOCKET_NETWORK socket28   6527 driver1d
SOCKET_NETWORK socket29   6528 driver1e
SOCKET_NETWORK socket30   6529 driver1f
SOCKET_NETWORK socket31   6530 driver1a
SOCKET_NETWORK socket32   6531 driver1b
SOCKET_NETWORK socket33   6532 driver1c
SOCKET_NETWORK socket34   6533 driver1d
SOCKET_NETWORK socket35   6534 driver1e
SOCKET_NETWORK socket36   6535 driver1f
SOCKET_NETWORK socket37   6536 driver1a
SOCKET_NETWORK socket38   6537 driver1b
SOCKET_NETWORK socket39   6538 driver1c
SOCKET_NETWORK socket40   6539 driver1d
SOCKET_NETWORK socket41   6540 driver1e
SOCKET_NETWORK socket42   6541 driver1f
SOCKET_NETWORK socket43   6542 driver1a
SOCKET_NETWORK socket44   6543 driver1b
SOCKET_NETWORK socket45   6544 driver1c
SOCKET_NETWORK socket46   6545 driver1d
SOCKET_NETWORK socket47   6546 driver1e
SOCKET_NETWORK socket48   6547 driver1f
SOCKET_NETWORK socket49   6548 driver1a
SOCKET_NETWORK socket50   6549 driver1b
SOCKET_NETWORK socket51   6550 driver1c
SOCKET_NETWORK socket52   6551 driver1d
SOCKET_NETWORK socket53   6552 driver1e
SOCKET_NETWORK socket54   6553 driver1f
SOCKET_NETWORK socket55   6554 driver1a
SOCKET_NETWORK socket56   6555 driver1b
SOCKET_NETWORK socket57   6556 driver2a
SOCKET_NETWORK socket58   6557 driver2b
SOCKET_NETWORK socket59   6558 driver2c
SOCKET_NETWORK socket60   6559 driver2d
```

SOCKET_NETWORK socket61 6560 driver2e
SOCKET_NETWORK socket62 6561 driver2f
SOCKET_NETWORK socket63 6562 driver2a
SOCKET_NETWORK socket64 6563 driver2b
SOCKET_NETWORK socket65 6564 driver2c
SOCKET_NETWORK socket66 6565 driver2d
SOCKET_NETWORK socket67 6566 driver2e
SOCKET_NETWORK socket68 6567 driver2f
SOCKET_NETWORK socket69 6568 driver2a
SOCKET_NETWORK socket70 6569 driver2b
SOCKET_NETWORK socket71 6570 driver2c
SOCKET_NETWORK socket72 6571 driver2d
SOCKET_NETWORK socket73 6572 driver2e
SOCKET_NETWORK socket74 6573 driver2f
SOCKET_NETWORK socket75 6574 driver2a
SOCKET_NETWORK socket76 6575 driver2b
SOCKET_NETWORK socket77 6576 driver2c
SOCKET_NETWORK socket78 6577 driver2d
SOCKET_NETWORK socket79 6578 driver2e
SOCKET_NETWORK socket80 6579 driver2f
SOCKET_NETWORK socket81 6580 driver2a
SOCKET_NETWORK socket82 6581 driver2b
SOCKET_NETWORK socket83 6582 driver2c
SOCKET_NETWORK socket84 6583 driver2d
SOCKET_NETWORK socket85 6584 driver2e
SOCKET_NETWORK socket86 6585 driver2f
SOCKET_NETWORK socket87 6586 driver2a
SOCKET_NETWORK socket88 6587 driver2b
SOCKET_NETWORK socket89 6588 driver2c
SOCKET_NETWORK socket90 6589 driver2d
SOCKET_NETWORK socket91 6590 driver2e
SOCKET_NETWORK socket92 6591 driver2f
SOCKET_NETWORK socket93 6592 driver2a
SOCKET_NETWORK socket94 6593 driver2b
SOCKET_NETWORK socket95 6594 driver2c
SOCKET_NETWORK socket96 6595 driver2d
SOCKET_NETWORK socket97 6596 driver2e
SOCKET_NETWORK socket98 6597 driver2f
SOCKET_NETWORK socket99 6598 driver2a
SOCKET_NETWORK socket100 6599 driver2b
SOCKET_NETWORK socket101 6600 driver2c
SOCKET_NETWORK socket102 6601 driver2d
SOCKET_NETWORK socket103 6602 driver2e
SOCKET_NETWORK socket104 6603 driver2f
SOCKET_NETWORK socket105 6604 driver2a
SOCKET_NETWORK socket106 6605 driver2b
SOCKET_NETWORK socket107 6606 driver2c
SOCKET_NETWORK socket108 6607 driver2d
SOCKET_NETWORK socket109 6608 driver2e
SOCKET_NETWORK socket110 6609 driver2f
SOCKET_NETWORK socket111 6610 driver2a
SOCKET_NETWORK socket112 6611 driver2b
SOCKET_NETWORK socket113 6500 driver3a
SOCKET_NETWORK socket114 6501 driver3b
SOCKET_NETWORK socket115 6502 driver3c
SOCKET_NETWORK socket116 6503 driver3d
SOCKET_NETWORK socket117 6504 driver3e
SOCKET_NETWORK socket118 6505 driver3f
SOCKET_NETWORK socket119 6506 driver3a
SOCKET_NETWORK socket120 6507 driver3b
SOCKET_NETWORK socket121 6508 driver3c
SOCKET_NETWORK socket122 6509 driver3d
SOCKET_NETWORK socket123 6510 driver3e
SOCKET_NETWORK socket124 6511 driver3f
SOCKET_NETWORK socket125 6512 driver3a
SOCKET_NETWORK socket126 6513 driver3b
SOCKET_NETWORK socket127 6514 driver3c
SOCKET_NETWORK socket128 6515 driver3d
SOCKET_NETWORK socket129 6516 driver3e
SOCKET_NETWORK socket130 6517 driver3f
SOCKET_NETWORK socket131 6518 driver3a
SOCKET_NETWORK socket132 6519 driver3b
SOCKET_NETWORK socket133 6520 driver3c
SOCKET_NETWORK socket134 6521 driver3d
SOCKET_NETWORK socket135 6522 driver3e
SOCKET_NETWORK socket136 6523 driver3f
SOCKET_NETWORK socket137 6524 driver3a
SOCKET_NETWORK socket138 6525 driver3b
SOCKET_NETWORK socket139 6526 driver3c
SOCKET_NETWORK socket140 6527 driver3d
SOCKET_NETWORK socket141 6528 driver3e
SOCKET_NETWORK socket142 6529 driver3f
SOCKET_NETWORK socket143 6530 driver3a
SOCKET_NETWORK socket144 6531 driver3b
SOCKET_NETWORK socket145 6532 driver3c
SOCKET_NETWORK socket146 6533 driver3d
SOCKET_NETWORK socket147 6534 driver3e
SOCKET_NETWORK socket148 6535 driver3f
SOCKET_NETWORK socket149 6536 driver3a
SOCKET_NETWORK socket150 6537 driver3b
SOCKET_NETWORK socket151 6538 driver3c
SOCKET_NETWORK socket152 6539 driver3d
SOCKET_NETWORK socket153 6540 driver3e
SOCKET_NETWORK socket154 6541 driver3f
SOCKET_NETWORK socket155 6542 driver3a
SOCKET_NETWORK socket156 6543 driver3b
SOCKET_NETWORK socket157 6544 driver3c
SOCKET_NETWORK socket158 6545 driver3d
SOCKET_NETWORK socket159 6546 driver3e
SOCKET_NETWORK socket160 6547 driver3f
SOCKET_NETWORK socket161 6548 driver3a
SOCKET_NETWORK socket162 6549 driver3b
SOCKET_NETWORK socket163 6550 driver3c
SOCKET_NETWORK socket164 6551 driver3d
SOCKET_NETWORK socket165 6552 driver3e
SOCKET_NETWORK socket166 6553 driver3f
SOCKET_NETWORK socket167 6554 driver3a
SOCKET_NETWORK socket168 6555 driver3b
SOCKET_NETWORK socket169 6556 driver4a
SOCKET_NETWORK socket170 6557 driver4b
SOCKET_NETWORK socket171 6558 driver4c
SOCKET_NETWORK socket172 6559 driver4d
SOCKET_NETWORK socket173 6560 driver4e
SOCKET_NETWORK socket174 6561 driver4f
SOCKET_NETWORK socket175 6562 driver4a
SOCKET_NETWORK socket176 6563 driver4b
SOCKET_NETWORK socket177 6564 driver4c
SOCKET_NETWORK socket178 6565 driver4d
SOCKET_NETWORK socket179 6566 driver4e
SOCKET_NETWORK socket180 6567 driver4f
SOCKET_NETWORK socket181 6568 driver4a
SOCKET_NETWORK socket182 6569 driver4b
SOCKET_NETWORK socket183 6570 driver4c
SOCKET_NETWORK socket184 6571 driver4d
SOCKET_NETWORK socket185 6572 driver4e
SOCKET_NETWORK socket186 6573 driver4f
SOCKET_NETWORK socket187 6574 driver4a
SOCKET_NETWORK socket188 6575 driver4b
SOCKET_NETWORK socket189 6576 driver4c
SOCKET_NETWORK socket190 6577 driver4d
SOCKET_NETWORK socket191 6578 driver4e
SOCKET_NETWORK socket192 6579 driver4f
SOCKET_NETWORK socket193 6580 driver4a
SOCKET_NETWORK socket194 6581 driver4b
SOCKET_NETWORK socket195 6582 driver4c
SOCKET_NETWORK socket196 6583 driver4d
SOCKET_NETWORK socket197 6584 driver4e
SOCKET_NETWORK socket198 6585 driver4f
SOCKET_NETWORK socket199 6586 driver4a
SOCKET_NETWORK socket200 6587 driver4b
SOCKET_NETWORK socket201 6588 driver4c
SOCKET_NETWORK socket202 6589 driver4d
SOCKET_NETWORK socket203 6590 driver4e
SOCKET_NETWORK socket204 6591 driver4f
SOCKET_NETWORK socket205 6592 driver4a
SOCKET_NETWORK socket206 6593 driver4b
SOCKET_NETWORK socket207 6594 driver4c
SOCKET_NETWORK socket208 6595 driver4d
SOCKET_NETWORK socket209 6596 driver4e
SOCKET_NETWORK socket210 6597 driver4f
SOCKET_NETWORK socket211 6598 driver4a
SOCKET_NETWORK socket212 6599 driver4b
SOCKET_NETWORK socket213 6600 driver4c
SOCKET_NETWORK socket214 6601 driver4d
SOCKET_NETWORK socket215 6602 driver4e
SOCKET_NETWORK socket216 6603 driver4f
SOCKET_NETWORK socket217 6604 driver4a
SOCKET_NETWORK socket218 6605 driver4b
SOCKET_NETWORK socket219 6606 driver4c
SOCKET_NETWORK socket220 6607 driver4d
SOCKET_NETWORK socket221 6608 driver4e
SOCKET_NETWORK socket222 6609 driver4f
SOCKET_NETWORK socket223 6610 driver4a
SOCKET_NETWORK socket224 6611 driver4b
SOCKET_NETWORK socket225 6500 driver5a
SOCKET_NETWORK socket226 6501 driver5b
SOCKET_NETWORK socket227 6502 driver5c
SOCKET_NETWORK socket228 6503 driver5d
SOCKET_NETWORK socket229 6504 driver5e
SOCKET_NETWORK socket230 6505 driver5f
SOCKET_NETWORK socket231 6506 driver5a
SOCKET_NETWORK socket232 6507 driver5b
SOCKET_NETWORK socket233 6508 driver5c
SOCKET_NETWORK socket234 6509 driver5d
SOCKET_NETWORK socket235 6510 driver5e
SOCKET_NETWORK socket236 6511 driver5f
SOCKET_NETWORK socket237 6512 driver5a
SOCKET_NETWORK socket238 6513 driver5b
SOCKET_NETWORK socket239 6514 driver5c
SOCKET_NETWORK socket240 6515 driver5d
SOCKET_NETWORK socket241 6516 driver5e
SOCKET_NETWORK socket242 6517 driver5f
SOCKET_NETWORK socket243 6518 driver5a
SOCKET_NETWORK socket244 6519 driver5b
SOCKET_NETWORK socket245 6520 driver5c
SOCKET_NETWORK socket246 6521 driver5d
SOCKET_NETWORK socket247 6522 driver5e
SOCKET_NETWORK socket248 6523 driver5f
SOCKET_NETWORK socket249 6524 driver5a
SOCKET_NETWORK socket250 6525 driver5b
SOCKET_NETWORK socket251 6526 driver5c
SOCKET_NETWORK socket252 6527 driver5d
SOCKET_NETWORK socket253 6528 driver5e
SOCKET_NETWORK socket254 6529 driver5f
SOCKET_NETWORK socket255 6530 driver5a
SOCKET_NETWORK socket256 6531 driver5b
SOCKET_NETWORK socket257 6532 driver5c
SOCKET_NETWORK socket258 6533 driver5d

SOCKET_NETWORK socket259 6534 driver5e
SOCKET_NETWORK socket260 6535 driver5f
SOCKET_NETWORK socket261 6536 driver5a
SOCKET_NETWORK socket262 6537 driver5b
SOCKET_NETWORK socket263 6538 driver5c
SOCKET_NETWORK socket264 6539 driver5d
SOCKET_NETWORK socket265 6540 driver5e
SOCKET_NETWORK socket266 6541 driver5f
SOCKET_NETWORK socket267 6542 driver5a
SOCKET_NETWORK socket268 6543 driver5b
SOCKET_NETWORK socket269 6544 driver5c
SOCKET_NETWORK socket270 6545 driver5d
SOCKET_NETWORK socket271 6546 driver5e
SOCKET_NETWORK socket272 6547 driver5f
SOCKET_NETWORK socket273 6548 driver5a
SOCKET_NETWORK socket274 6549 driver5b
SOCKET_NETWORK socket275 6550 driver5c
SOCKET_NETWORK socket276 6551 driver5d
SOCKET_NETWORK socket277 6552 driver5e
SOCKET_NETWORK socket278 6553 driver5f
SOCKET_NETWORK socket279 6554 driver5a
SOCKET_NETWORK socket280 6555 driver5b
SOCKET_NETWORK socket281 6556 driver6a
SOCKET_NETWORK socket282 6557 driver6b
SOCKET_NETWORK socket283 6558 driver6c
SOCKET_NETWORK socket284 6559 driver6d
SOCKET_NETWORK socket285 6560 driver6e
SOCKET_NETWORK socket286 6561 driver6f
SOCKET_NETWORK socket287 6562 driver6a
SOCKET_NETWORK socket288 6563 driver6b
SOCKET_NETWORK socket289 6564 driver6c
SOCKET_NETWORK socket290 6565 driver6d
SOCKET_NETWORK socket291 6566 driver6e
SOCKET_NETWORK socket292 6567 driver6f
SOCKET_NETWORK socket293 6568 driver6a
SOCKET_NETWORK socket294 6569 driver6b
SOCKET_NETWORK socket295 6570 driver6c
SOCKET_NETWORK socket296 6571 driver6d
SOCKET_NETWORK socket297 6572 driver6e
SOCKET_NETWORK socket298 6573 driver6f
SOCKET_NETWORK socket299 6574 driver6a
SOCKET_NETWORK socket300 6575 driver6b
SOCKET_NETWORK socket301 6576 driver6c
SOCKET_NETWORK socket302 6577 driver6d
SOCKET_NETWORK socket303 6578 driver6e
SOCKET_NETWORK socket304 6579 driver6f
SOCKET_NETWORK socket305 6580 driver6a
SOCKET_NETWORK socket306 6581 driver6b
SOCKET_NETWORK socket307 6582 driver6c
SOCKET_NETWORK socket308 6583 driver6d
SOCKET_NETWORK socket309 6584 driver6e
SOCKET_NETWORK socket310 6585 driver6f
SOCKET_NETWORK socket311 6586 driver6a
SOCKET_NETWORK socket312 6587 driver6b
SOCKET_NETWORK socket313 6588 driver6c
SOCKET_NETWORK socket314 6589 driver6d
SOCKET_NETWORK socket315 6590 driver6e
SOCKET_NETWORK socket316 6591 driver6f
SOCKET_NETWORK socket317 6592 driver6a
SOCKET_NETWORK socket318 6593 driver6b
SOCKET_NETWORK socket319 6594 driver6c
SOCKET_NETWORK socket320 6595 driver6d
SOCKET_NETWORK socket321 6596 driver6e
SOCKET_NETWORK socket322 6597 driver6f
SOCKET_NETWORK socket323 6598 driver6a
SOCKET_NETWORK socket324 6599 driver6b
SOCKET_NETWORK socket325 6600 driver6c
SOCKET_NETWORK socket326 6601 driver6d
SOCKET_NETWORK socket327 6602 driver6e
SOCKET_NETWORK socket328 6603 driver6f
SOCKET_NETWORK socket329 6604 driver6a
SOCKET_NETWORK socket330 6605 driver6b
SOCKET_NETWORK socket331 6606 driver6c
SOCKET_NETWORK socket332 6607 driver6d
SOCKET_NETWORK socket333 6608 driver6e
SOCKET_NETWORK socket334 6609 driver6f
SOCKET_NETWORK socket335 6610 driver6a
SOCKET_NETWORK socket336 6611 driver6b
#elif MASTER_NUM2
SOCKET_NETWORK socket337 6500 driver7a
SOCKET_NETWORK socket338 6501 driver7b
SOCKET_NETWORK socket339 6502 driver7c
SOCKET_NETWORK socket340 6503 driver7d
SOCKET_NETWORK socket341 6504 driver7e
SOCKET_NETWORK socket342 6505 driver7f
SOCKET_NETWORK socket343 6506 driver7a
SOCKET_NETWORK socket344 6507 driver7b
SOCKET_NETWORK socket345 6508 driver7c
SOCKET_NETWORK socket346 6509 driver7d
SOCKET_NETWORK socket347 6510 driver7e
SOCKET_NETWORK socket348 6511 driver7f
SOCKET_NETWORK socket349 6512 driver7a
SOCKET_NETWORK socket350 6513 driver7b
SOCKET_NETWORK socket351 6514 driver7c
SOCKET_NETWORK socket352 6515 driver7d
SOCKET_NETWORK socket353 6516 driver7e
SOCKET_NETWORK socket354 6517 driver7f
SOCKET_NETWORK socket355 6518 driver7a
SOCKET_NETWORK socket356 6519 driver7b
SOCKET_NETWORK socket357 6520 driver7c
SOCKET_NETWORK socket358 6521 driver7d
SOCKET_NETWORK socket359 6522 driver7e
SOCKET_NETWORK socket360 6523 driver7f
SOCKET_NETWORK socket361 6524 driver7a
SOCKET_NETWORK socket362 6525 driver7b
SOCKET_NETWORK socket363 6526 driver7c
SOCKET_NETWORK socket364 6527 driver7d
SOCKET_NETWORK socket365 6528 driver7e
SOCKET_NETWORK socket366 6529 driver7f
SOCKET_NETWORK socket367 6530 driver7a
SOCKET_NETWORK socket368 6531 driver7b
SOCKET_NETWORK socket369 6532 driver7c
SOCKET_NETWORK socket370 6533 driver7d
SOCKET_NETWORK socket371 6534 driver7e
SOCKET_NETWORK socket372 6535 driver7f
SOCKET_NETWORK socket373 6536 driver7a
SOCKET_NETWORK socket374 6537 driver7b
SOCKET_NETWORK socket375 6538 driver7c
SOCKET_NETWORK socket376 6539 driver7d
SOCKET_NETWORK socket377 6540 driver7e
SOCKET_NETWORK socket378 6541 driver7f
SOCKET_NETWORK socket379 6542 driver7a
SOCKET_NETWORK socket380 6543 driver7b
SOCKET_NETWORK socket381 6544 driver7c
SOCKET_NETWORK socket382 6545 driver7d
SOCKET_NETWORK socket383 6546 driver7e
SOCKET_NETWORK socket384 6547 driver7f
SOCKET_NETWORK socket385 6548 driver7a
SOCKET_NETWORK socket386 6549 driver7b
SOCKET_NETWORK socket387 6550 driver7c
SOCKET_NETWORK socket388 6551 driver7d
SOCKET_NETWORK socket389 6552 driver7e
SOCKET_NETWORK socket390 6553 driver7f
SOCKET_NETWORK socket391 6554 driver7a
SOCKET_NETWORK socket392 6555 driver7b
SOCKET_NETWORK socket393 6556 driver8a
SOCKET_NETWORK socket394 6557 driver8b
SOCKET_NETWORK socket395 6558 driver8c
SOCKET_NETWORK socket396 6559 driver8d
SOCKET_NETWORK socket397 6560 driver8e
SOCKET_NETWORK socket398 6561 driver8f
SOCKET_NETWORK socket399 6562 driver8a
SOCKET_NETWORK socket400 6563 driver8b
SOCKET_NETWORK socket401 6564 driver8c
SOCKET_NETWORK socket402 6565 driver8d
SOCKET_NETWORK socket403 6566 driver8e
SOCKET_NETWORK socket404 6567 driver8f
SOCKET_NETWORK socket405 6568 driver8a
SOCKET_NETWORK socket406 6569 driver8b
SOCKET_NETWORK socket407 6570 driver8c
SOCKET_NETWORK socket408 6571 driver8d
SOCKET_NETWORK socket409 6572 driver8e
SOCKET_NETWORK socket410 6573 driver8f
SOCKET_NETWORK socket411 6574 driver8a
SOCKET_NETWORK socket412 6575 driver8b
SOCKET_NETWORK socket413 6576 driver8c
SOCKET_NETWORK socket414 6577 driver8d
SOCKET_NETWORK socket415 6578 driver8e
SOCKET_NETWORK socket416 6579 driver8f
SOCKET_NETWORK socket417 6580 driver8a
SOCKET_NETWORK socket418 6581 driver8b
SOCKET_NETWORK socket419 6582 driver8c
SOCKET_NETWORK socket420 6583 driver8d
SOCKET_NETWORK socket421 6584 driver8e
SOCKET_NETWORK socket422 6585 driver8f
SOCKET_NETWORK socket423 6586 driver8a
SOCKET_NETWORK socket424 6587 driver8b
SOCKET_NETWORK socket425 6588 driver8c
SOCKET_NETWORK socket426 6589 driver8d
SOCKET_NETWORK socket427 6590 driver8e
SOCKET_NETWORK socket428 6591 driver8f
SOCKET_NETWORK socket429 6592 driver8a
SOCKET_NETWORK socket430 6593 driver8b
SOCKET_NETWORK socket431 6594 driver8c
SOCKET_NETWORK socket432 6595 driver8d
SOCKET_NETWORK socket433 6596 driver8e
SOCKET_NETWORK socket434 6597 driver8f
SOCKET_NETWORK socket435 6598 driver8a
SOCKET_NETWORK socket436 6599 driver8b
SOCKET_NETWORK socket437 6600 driver8c
SOCKET_NETWORK socket438 6601 driver8d
SOCKET_NETWORK socket439 6602 driver8e
SOCKET_NETWORK socket440 6603 driver8f
SOCKET_NETWORK socket441 6604 driver8a
SOCKET_NETWORK socket442 6605 driver8b
SOCKET_NETWORK socket443 6606 driver8c
SOCKET_NETWORK socket444 6607 driver8d
SOCKET_NETWORK socket445 6608 driver8e
SOCKET_NETWORK socket446 6609 driver8f
SOCKET_NETWORK socket447 6610 driver8a
SOCKET_NETWORK socket448 6611 driver8b
SOCKET_NETWORK socket449 6500 driver9a
SOCKET_NETWORK socket450 6501 driver9b
SOCKET_NETWORK socket451 6502 driver9c
SOCKET_NETWORK socket452 6503 driver9d
SOCKET_NETWORK socket453 6504 driver9e
SOCKET_NETWORK socket454 6505 driver9f
SOCKET_NETWORK socket455 6506 driver9a

SOCKET_NETWORK socket456 6507 driver9b
SOCKET_NETWORK socket457 6508 driver9c
SOCKET_NETWORK socket458 6509 driver9d
SOCKET_NETWORK socket459 6510 driver9e
SOCKET_NETWORK socket460 6511 driver9f
SOCKET_NETWORK socket461 6512 driver9a
SOCKET_NETWORK socket462 6513 driver9b
SOCKET_NETWORK socket463 6514 driver9c
SOCKET_NETWORK socket464 6515 driver9d
SOCKET_NETWORK socket465 6516 driver9e
SOCKET_NETWORK socket466 6517 driver9f
SOCKET_NETWORK socket467 6518 driver9a
SOCKET_NETWORK socket468 6519 driver9b
SOCKET_NETWORK socket469 6520 driver9c
SOCKET_NETWORK socket470 6521 driver9d
SOCKET_NETWORK socket471 6522 driver9e
SOCKET_NETWORK socket472 6523 driver9f
SOCKET_NETWORK socket473 6524 driver9a
SOCKET_NETWORK socket474 6525 driver9b
SOCKET_NETWORK socket475 6526 driver9c
SOCKET_NETWORK socket476 6527 driver9d
SOCKET_NETWORK socket477 6528 driver9e
SOCKET_NETWORK socket478 6529 driver9f
SOCKET_NETWORK socket479 6530 driver9a
SOCKET_NETWORK socket480 6531 driver9b
SOCKET_NETWORK socket481 6532 driver9c
SOCKET_NETWORK socket482 6533 driver9d
SOCKET_NETWORK socket483 6534 driver9e
SOCKET_NETWORK socket484 6535 driver9f
SOCKET_NETWORK socket485 6536 driver9a
SOCKET_NETWORK socket486 6537 driver9b
SOCKET_NETWORK socket487 6538 driver9c
SOCKET_NETWORK socket488 6539 driver9d
SOCKET_NETWORK socket489 6540 driver9e
SOCKET_NETWORK socket490 6541 driver9f
SOCKET_NETWORK socket491 6542 driver9a
SOCKET_NETWORK socket492 6543 driver9b
SOCKET_NETWORK socket493 6544 driver9c
SOCKET_NETWORK socket494 6545 driver9d
SOCKET_NETWORK socket495 6546 driver9e
SOCKET_NETWORK socket496 6547 driver9f
SOCKET_NETWORK socket497 6548 driver9a
SOCKET_NETWORK socket498 6549 driver9b
SOCKET_NETWORK socket499 6550 driver9c
SOCKET_NETWORK socket500 6551 driver9d
SOCKET_NETWORK socket501 6552 driver9e
SOCKET_NETWORK socket502 6553 driver9f
SOCKET_NETWORK socket503 6554 driver9a
SOCKET_NETWORK socket504 6555 driver9b
SOCKET_NETWORK socket505 6556 driver10a
SOCKET_NETWORK socket506 6557 driver10b
SOCKET_NETWORK socket507 6558 driver10c
SOCKET_NETWORK socket508 6559 driver10d
SOCKET_NETWORK socket509 6560 driver10e
SOCKET_NETWORK socket510 6561 driver10f
SOCKET_NETWORK socket511 6562 driver10a
SOCKET_NETWORK socket512 6563 driver10b
SOCKET_NETWORK socket513 6564 driver10c
SOCKET_NETWORK socket514 6565 driver10d
SOCKET_NETWORK socket515 6566 driver10e
SOCKET_NETWORK socket516 6567 driver10f
SOCKET_NETWORK socket517 6568 driver10a
SOCKET_NETWORK socket518 6569 driver10b
SOCKET_NETWORK socket519 6570 driver10c
SOCKET_NETWORK socket520 6571 driver10d
SOCKET_NETWORK socket521 6572 driver10e
SOCKET_NETWORK socket522 6573 driver10f
SOCKET_NETWORK socket523 6574 driver10a
SOCKET_NETWORK socket524 6575 driver10b
SOCKET_NETWORK socket525 6576 driver10c
SOCKET_NETWORK socket526 6577 driver10d
SOCKET_NETWORK socket527 6578 driver10e
SOCKET_NETWORK socket528 6579 driver10f
SOCKET_NETWORK socket529 6580 driver10a
SOCKET_NETWORK socket530 6581 driver10b
SOCKET_NETWORK socket531 6582 driver10c
SOCKET_NETWORK socket532 6583 driver10d
SOCKET_NETWORK socket533 6584 driver10e
SOCKET_NETWORK socket534 6585 driver10f
SOCKET_NETWORK socket535 6586 driver10a
SOCKET_NETWORK socket536 6587 driver10b
SOCKET_NETWORK socket537 6588 driver10c
SOCKET_NETWORK socket538 6589 driver10d
SOCKET_NETWORK socket539 6590 driver10e
SOCKET_NETWORK socket540 6591 driver10f
SOCKET_NETWORK socket541 6592 driver10a
SOCKET_NETWORK socket542 6593 driver10b
SOCKET_NETWORK socket543 6594 driver10c
SOCKET_NETWORK socket544 6595 driver10d
SOCKET_NETWORK socket545 6596 driver10e
SOCKET_NETWORK socket546 6597 driver10f
SOCKET_NETWORK socket547 6598 driver10a
SOCKET_NETWORK socket548 6599 driver10b
SOCKET_NETWORK socket549 6600 driver10c
SOCKET_NETWORK socket550 6601 driver10d
SOCKET_NETWORK socket551 6602 driver10e
SOCKET_NETWORK socket552 6603 driver10f
SOCKET_NETWORK socket553 6604 driver10a
SOCKET_NETWORK socket554 6605 driver10b
SOCKET_NETWORK socket555 6606 driver10c
SOCKET_NETWORK socket556 6607 driver10d
SOCKET_NETWORK socket557 6608 driver10e
SOCKET_NETWORK socket558 6609 driver10f
SOCKET_NETWORK socket559 6610 driver10a
SOCKET_NETWORK socket560 6611 driver10b
SOCKET_NETWORK socket561 6500 driver11a
SOCKET_NETWORK socket562 6501 driver11b
SOCKET_NETWORK socket563 6502 driver11c
SOCKET_NETWORK socket564 6503 driver11d
SOCKET_NETWORK socket565 6504 driver11e
SOCKET_NETWORK socket566 6505 driver11f
SOCKET_NETWORK socket567 6506 driver11a
SOCKET_NETWORK socket568 6507 driver11b
SOCKET_NETWORK socket569 6508 driver11c
SOCKET_NETWORK socket570 6509 driver11d
SOCKET_NETWORK socket571 6510 driver11e
SOCKET_NETWORK socket572 6511 driver11f
SOCKET_NETWORK socket573 6512 driver11a
SOCKET_NETWORK socket574 6513 driver11b
SOCKET_NETWORK socket575 6514 driver11c
SOCKET_NETWORK socket576 6515 driver11d
SOCKET_NETWORK socket577 6516 driver11e
SOCKET_NETWORK socket578 6517 driver11f
SOCKET_NETWORK socket579 6518 driver11a
SOCKET_NETWORK socket580 6519 driver11b
SOCKET_NETWORK socket581 6520 driver11c
SOCKET_NETWORK socket582 6521 driver11d
SOCKET_NETWORK socket583 6522 driver11e
SOCKET_NETWORK socket584 6523 driver11f
SOCKET_NETWORK socket585 6524 driver11a
SOCKET_NETWORK socket586 6525 driver11b
SOCKET_NETWORK socket587 6526 driver11c
SOCKET_NETWORK socket588 6527 driver11d
SOCKET_NETWORK socket589 6528 driver11e
SOCKET_NETWORK socket590 6529 driver11f
SOCKET_NETWORK socket591 6530 driver11a
SOCKET_NETWORK socket592 6531 driver11b
SOCKET_NETWORK socket593 6532 driver11c
SOCKET_NETWORK socket594 6533 driver11d
SOCKET_NETWORK socket595 6534 driver11e
SOCKET_NETWORK socket596 6535 driver11f
SOCKET_NETWORK socket597 6536 driver11a
SOCKET_NETWORK socket598 6537 driver11b
SOCKET_NETWORK socket599 6538 driver11c
SOCKET_NETWORK socket600 6539 driver11d
SOCKET_NETWORK socket601 6540 driver11e
SOCKET_NETWORK socket602 6541 driver11f
SOCKET_NETWORK socket603 6542 driver11a
SOCKET_NETWORK socket604 6543 driver11b
SOCKET_NETWORK socket605 6544 driver11c
SOCKET_NETWORK socket606 6545 driver11d
SOCKET_NETWORK socket607 6546 driver11e
SOCKET_NETWORK socket608 6547 driver11f
SOCKET_NETWORK socket609 6548 driver11a
SOCKET_NETWORK socket610 6549 driver11b
SOCKET_NETWORK socket611 6550 driver11c
SOCKET_NETWORK socket612 6551 driver11d
SOCKET_NETWORK socket613 6552 driver11e
SOCKET_NETWORK socket614 6553 driver11f
SOCKET_NETWORK socket615 6554 driver11a
SOCKET_NETWORK socket616 6555 driver11b
SOCKET_NETWORK socket617 6556 driver12a
SOCKET_NETWORK socket618 6557 driver12b
SOCKET_NETWORK socket619 6558 driver12c
SOCKET_NETWORK socket620 6559 driver12d
SOCKET_NETWORK socket621 6560 driver12e
SOCKET_NETWORK socket622 6561 driver12f
SOCKET_NETWORK socket623 6562 driver12a
SOCKET_NETWORK socket624 6563 driver12b
SOCKET_NETWORK socket625 6564 driver12c
SOCKET_NETWORK socket626 6565 driver12d
SOCKET_NETWORK socket627 6566 driver12e
SOCKET_NETWORK socket628 6567 driver12f
SOCKET_NETWORK socket629 6568 driver12a
SOCKET_NETWORK socket630 6569 driver12b
SOCKET_NETWORK socket631 6570 driver12c
SOCKET_NETWORK socket632 6571 driver12d
SOCKET_NETWORK socket633 6572 driver12e
SOCKET_NETWORK socket634 6573 driver12f
SOCKET_NETWORK socket635 6574 driver12a
SOCKET_NETWORK socket636 6575 driver12b
SOCKET_NETWORK socket637 6576 driver12c
SOCKET_NETWORK socket638 6577 driver12d
SOCKET_NETWORK socket639 6578 driver12e
SOCKET_NETWORK socket640 6579 driver12f
SOCKET_NETWORK socket641 6580 driver12a
SOCKET_NETWORK socket642 6581 driver12b
SOCKET_NETWORK socket643 6582 driver12c
SOCKET_NETWORK socket644 6583 driver12d
SOCKET_NETWORK socket645 6584 driver12e
SOCKET_NETWORK socket646 6585 driver12f
SOCKET_NETWORK socket647 6586 driver12a
SOCKET_NETWORK socket648 6587 driver12b
SOCKET_NETWORK socket649 6588 driver12c
SOCKET_NETWORK socket650 6589 driver12d
SOCKET_NETWORK socket651 6590 driver12e
SOCKET_NETWORK socket652 6591 driver12f
SOCKET_NETWORK socket653 6592 driver12a

SOCKET_NETWORK socket654 6593 driver12b
SOCKET_NETWORK socket655 6594 driver12c
SOCKET_NETWORK socket656 6595 driver12d
SOCKET_NETWORK socket657 6596 driver12e
SOCKET_NETWORK socket658 6597 driver12f
SOCKET_NETWORK socket659 6598 driver12a
SOCKET_NETWORK socket660 6599 driver12b
SOCKET_NETWORK socket661 6600 driver12c
SOCKET_NETWORK socket662 6601 driver12d
SOCKET_NETWORK socket663 6602 driver12e
SOCKET_NETWORK socket664 6603 driver12f
SOCKET_NETWORK socket665 6604 driver12a
SOCKET_NETWORK socket666 6605 driver12b
SOCKET_NETWORK socket667 6606 driver12c
SOCKET_NETWORK socket668 6607 driver12d
SOCKET_NETWORK socket669 6608 driver12e
SOCKET_NETWORK socket670 6609 driver12f
SOCKET_NETWORK socket671 6610 driver12a
SOCKET_NETWORK socket672 6611 driver12b
#elif MASTER_NUM3
SOCKET_NETWORK socket673 6500 driver13a
SOCKET_NETWORK socket674 6501 driver13b
SOCKET_NETWORK socket675 6502 driver13c
SOCKET_NETWORK socket676 6503 driver13d
SOCKET_NETWORK socket677 6504 driver13e
SOCKET_NETWORK socket678 6505 driver13f
SOCKET_NETWORK socket679 6506 driver13a
SOCKET_NETWORK socket680 6507 driver13b
SOCKET_NETWORK socket681 6508 driver13c
SOCKET_NETWORK socket682 6509 driver13d
SOCKET_NETWORK socket683 6510 driver13e
SOCKET_NETWORK socket684 6511 driver13f
SOCKET_NETWORK socket685 6512 driver13a
SOCKET_NETWORK socket686 6513 driver13b
SOCKET_NETWORK socket687 6514 driver13c
SOCKET_NETWORK socket688 6515 driver13d
SOCKET_NETWORK socket689 6516 driver13e
SOCKET_NETWORK socket690 6517 driver13f
SOCKET_NETWORK socket691 6518 driver13a
SOCKET_NETWORK socket692 6519 driver13b
SOCKET_NETWORK socket693 6520 driver13c
SOCKET_NETWORK socket694 6521 driver13d
SOCKET_NETWORK socket695 6522 driver13e
SOCKET_NETWORK socket696 6523 driver13f
SOCKET_NETWORK socket697 6524 driver13a
SOCKET_NETWORK socket698 6525 driver13b
SOCKET_NETWORK socket699 6526 driver13c
SOCKET_NETWORK socket700 6527 driver13d
SOCKET_NETWORK socket701 6528 driver13e
SOCKET_NETWORK socket702 6529 driver13f
SOCKET_NETWORK socket703 6530 driver13a
SOCKET_NETWORK socket704 6531 driver13b
SOCKET_NETWORK socket705 6532 driver13c
SOCKET_NETWORK socket706 6533 driver13d
SOCKET_NETWORK socket707 6534 driver13e
SOCKET_NETWORK socket708 6535 driver13f
SOCKET_NETWORK socket709 6536 driver13a
SOCKET_NETWORK socket710 6537 driver13b
SOCKET_NETWORK socket711 6538 driver13c
SOCKET_NETWORK socket712 6539 driver13d
SOCKET_NETWORK socket713 6540 driver13e
SOCKET_NETWORK socket714 6541 driver13f
SOCKET_NETWORK socket715 6542 driver13a
SOCKET_NETWORK socket716 6543 driver13b
SOCKET_NETWORK socket717 6544 driver13c
SOCKET_NETWORK socket718 6545 driver13d
SOCKET_NETWORK socket719 6546 driver13e
SOCKET_NETWORK socket720 6547 driver13f
SOCKET_NETWORK socket721 6548 driver13a
SOCKET_NETWORK socket722 6549 driver13b
SOCKET_NETWORK socket723 6550 driver13c
SOCKET_NETWORK socket724 6551 driver13d
SOCKET_NETWORK socket725 6552 driver13e
SOCKET_NETWORK socket726 6553 driver13f
SOCKET_NETWORK socket727 6554 driver13a
SOCKET_NETWORK socket728 6555 driver13b
SOCKET_NETWORK socket729 6556 driver14a
SOCKET_NETWORK socket730 6557 driver14b
SOCKET_NETWORK socket731 6558 driver14c
SOCKET_NETWORK socket732 6559 driver14d
SOCKET_NETWORK socket733 6560 driver14e
SOCKET_NETWORK socket734 6561 driver14f
SOCKET_NETWORK socket735 6562 driver14a
SOCKET_NETWORK socket736 6563 driver14b
SOCKET_NETWORK socket737 6564 driver14c
SOCKET_NETWORK socket738 6565 driver14d
SOCKET_NETWORK socket739 6566 driver14e
SOCKET_NETWORK socket740 6567 driver14f
SOCKET_NETWORK socket741 6568 driver14a
SOCKET_NETWORK socket742 6569 driver14b
SOCKET_NETWORK socket743 6570 driver14c
SOCKET_NETWORK socket744 6571 driver14d
SOCKET_NETWORK socket745 6572 driver14e
SOCKET_NETWORK socket746 6573 driver14f
SOCKET_NETWORK socket747 6574 driver14a
SOCKET_NETWORK socket748 6575 driver14b
SOCKET_NETWORK socket749 6576 driver14c
SOCKET_NETWORK socket750 6577 driver14d
SOCKET_NETWORK socket751 6578 driver14e
SOCKET_NETWORK socket752 6579 driver14f
SOCKET_NETWORK socket753 6580 driver14a
SOCKET_NETWORK socket754 6581 driver14b
SOCKET_NETWORK socket755 6582 driver14c
SOCKET_NETWORK socket756 6583 driver14d
SOCKET_NETWORK socket757 6584 driver14e
SOCKET_NETWORK socket758 6585 driver14f
SOCKET_NETWORK socket759 6586 driver14a
SOCKET_NETWORK socket760 6587 driver14b
SOCKET_NETWORK socket761 6588 driver14c
SOCKET_NETWORK socket762 6589 driver14d
SOCKET_NETWORK socket763 6590 driver14e
SOCKET_NETWORK socket764 6591 driver14f
SOCKET_NETWORK socket765 6592 driver14a
SOCKET_NETWORK socket766 6593 driver14b
SOCKET_NETWORK socket767 6594 driver14c
SOCKET_NETWORK socket768 6595 driver14d
SOCKET_NETWORK socket769 6596 driver14e
SOCKET_NETWORK socket770 6597 driver14f
SOCKET_NETWORK socket771 6598 driver14a
SOCKET_NETWORK socket772 6599 driver14b
SOCKET_NETWORK socket773 6600 driver14c
SOCKET_NETWORK socket774 6601 driver14d
SOCKET_NETWORK socket775 6602 driver14e
SOCKET_NETWORK socket776 6603 driver14f
SOCKET_NETWORK socket777 6604 driver14a
SOCKET_NETWORK socket778 6605 driver14b
SOCKET_NETWORK socket779 6606 driver14c
SOCKET_NETWORK socket780 6607 driver14d
SOCKET_NETWORK socket781 6608 driver14e
SOCKET_NETWORK socket782 6609 driver14f
SOCKET_NETWORK socket783 6610 driver14a
SOCKET_NETWORK socket784 6611 driver14b
SOCKET_NETWORK socket785 6500 driver15a
SOCKET_NETWORK socket786 6501 driver15b
SOCKET_NETWORK socket787 6502 driver15c
SOCKET_NETWORK socket788 6503 driver15d
SOCKET_NETWORK socket789 6504 driver15e
SOCKET_NETWORK socket790 6505 driver15f
SOCKET_NETWORK socket791 6506 driver15a
SOCKET_NETWORK socket792 6507 driver15b
SOCKET_NETWORK socket793 6508 driver15c
SOCKET_NETWORK socket794 6509 driver15d
SOCKET_NETWORK socket795 6510 driver15e
SOCKET_NETWORK socket796 6511 driver15f
SOCKET_NETWORK socket797 6512 driver15a
SOCKET_NETWORK socket798 6513 driver15b
SOCKET_NETWORK socket799 6514 driver15c
SOCKET_NETWORK socket800 6515 driver15d
SOCKET_NETWORK socket801 6516 driver15e
SOCKET_NETWORK socket802 6517 driver15f
SOCKET_NETWORK socket803 6518 driver15a
SOCKET_NETWORK socket804 6519 driver15b
SOCKET_NETWORK socket805 6520 driver15c
SOCKET_NETWORK socket806 6521 driver15d
SOCKET_NETWORK socket807 6522 driver15e
SOCKET_NETWORK socket808 6523 driver15f
SOCKET_NETWORK socket809 6524 driver15a
SOCKET_NETWORK socket810 6525 driver15b
SOCKET_NETWORK socket811 6526 driver15c
SOCKET_NETWORK socket812 6527 driver15d
SOCKET_NETWORK socket813 6528 driver15e
SOCKET_NETWORK socket814 6529 driver15f
SOCKET_NETWORK socket815 6530 driver15a
SOCKET_NETWORK socket816 6531 driver15b
SOCKET_NETWORK socket817 6532 driver15c
SOCKET_NETWORK socket818 6533 driver15d
SOCKET_NETWORK socket819 6534 driver15e
SOCKET_NETWORK socket820 6535 driver15f
SOCKET_NETWORK socket821 6536 driver15a
SOCKET_NETWORK socket822 6537 driver15b
SOCKET_NETWORK socket823 6538 driver15c
SOCKET_NETWORK socket824 6539 driver15d
SOCKET_NETWORK socket825 6540 driver15e
SOCKET_NETWORK socket826 6541 driver15f
SOCKET_NETWORK socket827 6542 driver15a
SOCKET_NETWORK socket828 6543 driver15b
SOCKET_NETWORK socket829 6544 driver15c
SOCKET_NETWORK socket830 6545 driver15d
SOCKET_NETWORK socket831 6546 driver15e
SOCKET_NETWORK socket832 6547 driver15f
SOCKET_NETWORK socket833 6548 driver15a
SOCKET_NETWORK socket834 6549 driver15b
SOCKET_NETWORK socket835 6550 driver15c
SOCKET_NETWORK socket836 6551 driver15d
SOCKET_NETWORK socket837 6552 driver15e
SOCKET_NETWORK socket838 6553 driver15f
SOCKET_NETWORK socket839 6554 driver15a
SOCKET_NETWORK socket840 6555 driver15b
SOCKET_NETWORK socket841 6556 driver16a
SOCKET_NETWORK socket842 6557 driver16b
SOCKET_NETWORK socket843 6558 driver16c
SOCKET_NETWORK socket844 6559 driver16d
SOCKET_NETWORK socket845 6560 driver16e
SOCKET_NETWORK socket846 6561 driver16f
SOCKET_NETWORK socket847 6562 driver16a
SOCKET_NETWORK socket848 6563 driver16b
SOCKET_NETWORK socket849 6564 driver16c
SOCKET_NETWORK socket850 6565 driver16d

```
SOCKET_NETWORK socket851  6566 driver16e
SOCKET_NETWORK socket852  6567 driver16f
SOCKET_NETWORK socket853  6568 driver16a
SOCKET_NETWORK socket854  6569 driver16b
SOCKET_NETWORK socket855  6570 driver16c
SOCKET_NETWORK socket856  6571 driver16d
SOCKET_NETWORK socket857  6572 driver16e
SOCKET_NETWORK socket858  6573 driver16f
SOCKET_NETWORK socket859  6574 driver16a
SOCKET_NETWORK socket860  6575 driver16b
SOCKET_NETWORK socket861  6576 driver16c
SOCKET_NETWORK socket862  6577 driver16d
SOCKET_NETWORK socket863  6578 driver16e
SOCKET_NETWORK socket864  6579 driver16f
SOCKET_NETWORK socket865  6580 driver16a
SOCKET_NETWORK socket866  6581 driver16b
SOCKET_NETWORK socket867  6582 driver16c
SOCKET_NETWORK socket868  6583 driver16d
SOCKET_NETWORK socket869  6584 driver16e
SOCKET_NETWORK socket870  6585 driver16f
SOCKET_NETWORK socket871  6586 driver16a
SOCKET_NETWORK socket872  6587 driver16b
SOCKET_NETWORK socket873  6588 driver16c
SOCKET_NETWORK socket874  6589 driver16d
SOCKET_NETWORK socket875  6590 driver16e
SOCKET_NETWORK socket876  6591 driver16f
SOCKET_NETWORK socket877  6592 driver16a
SOCKET_NETWORK socket878  6593 driver16b
SOCKET_NETWORK socket879  6594 driver16c
SOCKET_NETWORK socket880  6595 driver16d
SOCKET_NETWORK socket881  6596 driver16e
SOCKET_NETWORK socket882  6597 driver16f
SOCKET_NETWORK socket883  6598 driver16a
SOCKET_NETWORK socket884  6599 driver16b
SOCKET_NETWORK socket885  6600 driver16c
SOCKET_NETWORK socket886  6601 driver16d
SOCKET_NETWORK socket887  6602 driver16e
SOCKET_NETWORK socket888  6603 driver16f
SOCKET_NETWORK socket889  6604 driver16a
SOCKET_NETWORK socket890  6605 driver16b
SOCKET_NETWORK socket891  6606 driver16c
SOCKET_NETWORK socket892  6607 driver16d
SOCKET_NETWORK socket893  6608 driver16e
SOCKET_NETWORK socket894  6609 driver16f
SOCKET_NETWORK socket895  6610 driver16a
SOCKET_NETWORK socket896  6611 driver16b
SOCKET_NETWORK socket897  6500 driver17a
SOCKET_NETWORK socket898  6501 driver17b
SOCKET_NETWORK socket899  6502 driver17c
SOCKET_NETWORK socket900  6503 driver17d
SOCKET_NETWORK socket901  6504 driver17e
SOCKET_NETWORK socket902  6505 driver17f
SOCKET_NETWORK socket903  6506 driver17a
SOCKET_NETWORK socket904  6507 driver17b
SOCKET_NETWORK socket905  6508 driver17c
SOCKET_NETWORK socket906  6509 driver17d
SOCKET_NETWORK socket907  6510 driver17e
SOCKET_NETWORK socket908  6511 driver17f
SOCKET_NETWORK socket909  6512 driver17a
SOCKET_NETWORK socket910  6513 driver17b
SOCKET_NETWORK socket911  6514 driver17c
SOCKET_NETWORK socket912  6515 driver17d
SOCKET_NETWORK socket913  6516 driver17e
SOCKET_NETWORK socket914  6517 driver17f
SOCKET_NETWORK socket915  6518 driver17a
SOCKET_NETWORK socket916  6519 driver17b
SOCKET_NETWORK socket917  6520 driver17c
SOCKET_NETWORK socket918  6521 driver17d
SOCKET_NETWORK socket919  6522 driver17e
SOCKET_NETWORK socket920  6523 driver17f
SOCKET_NETWORK socket921  6524 driver17a
SOCKET_NETWORK socket922  6525 driver17b
SOCKET_NETWORK socket923  6526 driver17c
SOCKET_NETWORK socket924  6527 driver17d
SOCKET_NETWORK socket925  6528 driver17e
SOCKET_NETWORK socket926  6529 driver17f
SOCKET_NETWORK socket927  6530 driver17a
SOCKET_NETWORK socket928  6531 driver17b
SOCKET_NETWORK socket929  6532 driver17c
SOCKET_NETWORK socket930  6533 driver17d
SOCKET_NETWORK socket931  6534 driver17e
SOCKET_NETWORK socket932  6535 driver17f
SOCKET_NETWORK socket933  6536 driver17a
SOCKET_NETWORK socket934  6537 driver17b
SOCKET_NETWORK socket935  6538 driver17c
SOCKET_NETWORK socket936  6539 driver17d
SOCKET_NETWORK socket937  6540 driver17e
SOCKET_NETWORK socket938  6541 driver17f
SOCKET_NETWORK socket939  6542 driver17a
SOCKET_NETWORK socket940  6543 driver17b
SOCKET_NETWORK socket941  6544 driver17c
SOCKET_NETWORK socket942  6545 driver17d
SOCKET_NETWORK socket943  6546 driver17e
SOCKET_NETWORK socket944  6547 driver17f
SOCKET_NETWORK socket945  6548 driver17a
SOCKET_NETWORK socket946  6549 driver17b
SOCKET_NETWORK socket947  6550 driver17c
SOCKET_NETWORK socket948  6551 driver17d
SOCKET_NETWORK socket949  6552 driver17e
SOCKET_NETWORK socket950  6553 driver17f
SOCKET_NETWORK socket951  6554 driver17a
SOCKET_NETWORK socket952  6555 driver17b
SOCKET_NETWORK socket953  6556 driver18a
SOCKET_NETWORK socket954  6557 driver18b
SOCKET_NETWORK socket955  6558 driver18c
SOCKET_NETWORK socket956  6559 driver18d
SOCKET_NETWORK socket957  6560 driver18e
SOCKET_NETWORK socket958  6561 driver18f
SOCKET_NETWORK socket959  6562 driver18a
SOCKET_NETWORK socket960  6563 driver18b
SOCKET_NETWORK socket961  6564 driver18c
SOCKET_NETWORK socket962  6565 driver18d
SOCKET_NETWORK socket963  6566 driver18e
SOCKET_NETWORK socket964  6567 driver18f
SOCKET_NETWORK socket965  6568 driver18a
SOCKET_NETWORK socket966  6569 driver18b
SOCKET_NETWORK socket967  6570 driver18c
SOCKET_NETWORK socket968  6571 driver18d
SOCKET_NETWORK socket969  6572 driver18e
SOCKET_NETWORK socket970  6573 driver18f
SOCKET_NETWORK socket971  6574 driver18a
SOCKET_NETWORK socket972  6575 driver18b
SOCKET_NETWORK socket973  6576 driver18c
SOCKET_NETWORK socket974  6577 driver18d
SOCKET_NETWORK socket975  6578 driver18e
SOCKET_NETWORK socket976  6579 driver18f
SOCKET_NETWORK socket977  6580 driver18a
SOCKET_NETWORK socket978  6581 driver18b
SOCKET_NETWORK socket979  6582 driver18c
SOCKET_NETWORK socket980  6583 driver18d
SOCKET_NETWORK socket981  6584 driver18e
SOCKET_NETWORK socket982  6585 driver18f
SOCKET_NETWORK socket983  6586 driver18a
SOCKET_NETWORK socket984  6587 driver18b
SOCKET_NETWORK socket985  6588 driver18c
SOCKET_NETWORK socket986  6589 driver18d
SOCKET_NETWORK socket987  6590 driver18e
SOCKET_NETWORK socket988  6591 driver18f
SOCKET_NETWORK socket989  6592 driver18a
SOCKET_NETWORK socket990  6593 driver18b
SOCKET_NETWORK socket991  6594 driver18c
SOCKET_NETWORK socket992  6595 driver18d
SOCKET_NETWORK socket993  6596 driver18e
SOCKET_NETWORK socket994  6597 driver18f
SOCKET_NETWORK socket995  6598 driver18a
SOCKET_NETWORK socket996  6599 driver18b
SOCKET_NETWORK socket997  6600 driver18c
SOCKET_NETWORK socket998  6601 driver18d
SOCKET_NETWORK socket999  6602 driver18e
SOCKET_NETWORK socket1000  6603 driver18f
SOCKET_NETWORK socket1001  6604 driver18a
SOCKET_NETWORK socket1002  6605 driver18b
SOCKET_NETWORK socket1003  6606 driver18c
SOCKET_NETWORK socket1004  6607 driver18d
SOCKET_NETWORK socket1005  6608 driver18e
SOCKET_NETWORK socket1006  6609 driver18f
SOCKET_NETWORK socket1007  6610 driver18a
SOCKET_NETWORK socket1008  6611 driver18b
#elif MASTER_NUM4
SOCKET_NETWORK socket1009  6500 driver19a
SOCKET_NETWORK socket1010  6501 driver19b
SOCKET_NETWORK socket1011  6502 driver19c
SOCKET_NETWORK socket1012  6503 driver19d
SOCKET_NETWORK socket1013  6504 driver19e
SOCKET_NETWORK socket1014  6505 driver19f
SOCKET_NETWORK socket1015  6506 driver19a
SOCKET_NETWORK socket1016  6507 driver19b
SOCKET_NETWORK socket1017  6508 driver19c
SOCKET_NETWORK socket1018  6509 driver19d
SOCKET_NETWORK socket1019  6510 driver19e
SOCKET_NETWORK socket1020  6511 driver19f
SOCKET_NETWORK socket1021  6512 driver19a
SOCKET_NETWORK socket1022  6513 driver19b
SOCKET_NETWORK socket1023  6514 driver19c
SOCKET_NETWORK socket1024  6515 driver19d
SOCKET_NETWORK socket1025  6516 driver19e
SOCKET_NETWORK socket1026  6517 driver19f
SOCKET_NETWORK socket1027  6518 driver19a
SOCKET_NETWORK socket1028  6519 driver19b
SOCKET_NETWORK socket1029  6520 driver19c
SOCKET_NETWORK socket1030  6521 driver19d
SOCKET_NETWORK socket1031  6522 driver19e
SOCKET_NETWORK socket1032  6523 driver19f
SOCKET_NETWORK socket1033  6524 driver19a
SOCKET_NETWORK socket1034  6525 driver19b
SOCKET_NETWORK socket1035  6526 driver19c
SOCKET_NETWORK socket1036  6527 driver19d
SOCKET_NETWORK socket1037  6528 driver19e
SOCKET_NETWORK socket1038  6529 driver19f
SOCKET_NETWORK socket1039  6530 driver19a
SOCKET_NETWORK socket1040  6531 driver19b
SOCKET_NETWORK socket1041  6532 driver19c
SOCKET_NETWORK socket1042  6533 driver19d
SOCKET_NETWORK socket1043  6534 driver19e
SOCKET_NETWORK socket1044  6535 driver19f
SOCKET_NETWORK socket1045  6536 driver19a
SOCKET_NETWORK socket1046  6537 driver19b
SOCKET_NETWORK socket1047  6538 driver19c
```

SOCKET_NETWORK socket1048 6539 driver19d
SOCKET_NETWORK socket1049 6540 driver19e
SOCKET_NETWORK socket1050 6541 driver19f
SOCKET_NETWORK socket1051 6542 driver19a
SOCKET_NETWORK socket1052 6543 driver19b
SOCKET_NETWORK socket1053 6544 driver19c
SOCKET_NETWORK socket1054 6545 driver19d
SOCKET_NETWORK socket1055 6546 driver19e
SOCKET_NETWORK socket1056 6547 driver19f
SOCKET_NETWORK socket1057 6548 driver19a
SOCKET_NETWORK socket1058 6549 driver19b
SOCKET_NETWORK socket1059 6550 driver19c
SOCKET_NETWORK socket1060 6551 driver19d
SOCKET_NETWORK socket1061 6552 driver19e
SOCKET_NETWORK socket1062 6553 driver19f
SOCKET_NETWORK socket1063 6554 driver19a
SOCKET_NETWORK socket1064 6555 driver19b
SOCKET_NETWORK socket1065 6556 driver20a
SOCKET_NETWORK socket1066 6557 driver20b
SOCKET_NETWORK socket1067 6558 driver20c
SOCKET_NETWORK socket1068 6559 driver20d
SOCKET_NETWORK socket1069 6560 driver20e
SOCKET_NETWORK socket1070 6561 driver20f
SOCKET_NETWORK socket1071 6562 driver20a
SOCKET_NETWORK socket1072 6563 driver20b
SOCKET_NETWORK socket1073 6564 driver20c
SOCKET_NETWORK socket1074 6565 driver20d
SOCKET_NETWORK socket1075 6566 driver20e
SOCKET_NETWORK socket1076 6567 driver20f
SOCKET_NETWORK socket1077 6568 driver20a
SOCKET_NETWORK socket1078 6569 driver20b
SOCKET_NETWORK socket1079 6570 driver20c
SOCKET_NETWORK socket1080 6571 driver20d
SOCKET_NETWORK socket1081 6572 driver20e
SOCKET_NETWORK socket1082 6573 driver20f
SOCKET_NETWORK socket1083 6574 driver20a
SOCKET_NETWORK socket1084 6575 driver20b
SOCKET_NETWORK socket1085 6576 driver20c
SOCKET_NETWORK socket1086 6577 driver20d
SOCKET_NETWORK socket1087 6578 driver20e
SOCKET_NETWORK socket1088 6579 driver20f
SOCKET_NETWORK socket1089 6580 driver20a
SOCKET_NETWORK socket1090 6581 driver20b
SOCKET_NETWORK socket1091 6582 driver20c
SOCKET_NETWORK socket1092 6583 driver20d
SOCKET_NETWORK socket1093 6584 driver20e
SOCKET_NETWORK socket1094 6585 driver20f
SOCKET_NETWORK socket1095 6586 driver20a
SOCKET_NETWORK socket1096 6587 driver20b
SOCKET_NETWORK socket1097 6588 driver20c
SOCKET_NETWORK socket1098 6589 driver20d
SOCKET_NETWORK socket1099 6590 driver20e
SOCKET_NETWORK socket1100 6591 driver20f
SOCKET_NETWORK socket1101 6592 driver20a
SOCKET_NETWORK socket1102 6593 driver20b
SOCKET_NETWORK socket1103 6594 driver20c
SOCKET_NETWORK socket1104 6595 driver20d
SOCKET_NETWORK socket1105 6596 driver20e
SOCKET_NETWORK socket1106 6597 driver20f
SOCKET_NETWORK socket1107 6598 driver20a
SOCKET_NETWORK socket1108 6599 driver20b
SOCKET_NETWORK socket1109 6600 driver20c
SOCKET_NETWORK socket1110 6601 driver20d
SOCKET_NETWORK socket1111 6602 driver20e
SOCKET_NETWORK socket1112 6603 driver20f
SOCKET_NETWORK socket1113 6604 driver20a
SOCKET_NETWORK socket1114 6605 driver20b
SOCKET_NETWORK socket1115 6606 driver20c
SOCKET_NETWORK socket1116 6607 driver20d
SOCKET_NETWORK socket1117 6608 driver20e
SOCKET_NETWORK socket1118 6609 driver20f
SOCKET_NETWORK socket1119 6610 driver20a
SOCKET_NETWORK socket1120 6611 driver20b
SOCKET_NETWORK socket1121 6500 driver21a
SOCKET_NETWORK socket1122 6501 driver21b
SOCKET_NETWORK socket1123 6502 driver21c
SOCKET_NETWORK socket1124 6503 driver21d
SOCKET_NETWORK socket1125 6504 driver21e
SOCKET_NETWORK socket1126 6505 driver21f
SOCKET_NETWORK socket1127 6506 driver21a
SOCKET_NETWORK socket1128 6507 driver21b
SOCKET_NETWORK socket1129 6508 driver21c
SOCKET_NETWORK socket1130 6509 driver21d
SOCKET_NETWORK socket1131 6510 driver21e
SOCKET_NETWORK socket1132 6511 driver21f
SOCKET_NETWORK socket1133 6512 driver21a
SOCKET_NETWORK socket1134 6513 driver21b
SOCKET_NETWORK socket1135 6514 driver21c
SOCKET_NETWORK socket1136 6515 driver21d
SOCKET_NETWORK socket1137 6516 driver21e
SOCKET_NETWORK socket1138 6517 driver21f
SOCKET_NETWORK socket1139 6518 driver21a
SOCKET_NETWORK socket1140 6519 driver21b
SOCKET_NETWORK socket1141 6520 driver21c
SOCKET_NETWORK socket1142 6521 driver21d
SOCKET_NETWORK socket1143 6522 driver21e
SOCKET_NETWORK socket1144 6523 driver21f
SOCKET_NETWORK socket1145 6524 driver21a
SOCKET_NETWORK socket1146 6525 driver21b
SOCKET_NETWORK socket1147 6526 driver21c
SOCKET_NETWORK socket1148 6527 driver21d
SOCKET_NETWORK socket1149 6528 driver21e
SOCKET_NETWORK socket1150 6529 driver21f
SOCKET_NETWORK socket1151 6530 driver21a
SOCKET_NETWORK socket1152 6531 driver21b
SOCKET_NETWORK socket1153 6532 driver21c
SOCKET_NETWORK socket1154 6533 driver21d
SOCKET_NETWORK socket1155 6534 driver21e
SOCKET_NETWORK socket1156 6535 driver21f
SOCKET_NETWORK socket1157 6536 driver21a
SOCKET_NETWORK socket1158 6537 driver21b
SOCKET_NETWORK socket1159 6538 driver21c
SOCKET_NETWORK socket1160 6539 driver21d
SOCKET_NETWORK socket1161 6540 driver21e
SOCKET_NETWORK socket1162 6541 driver21f
SOCKET_NETWORK socket1163 6542 driver21a
SOCKET_NETWORK socket1164 6543 driver21b
SOCKET_NETWORK socket1165 6544 driver21c
SOCKET_NETWORK socket1166 6545 driver21d
SOCKET_NETWORK socket1167 6546 driver21e
SOCKET_NETWORK socket1168 6547 driver21f
SOCKET_NETWORK socket1169 6548 driver21a
SOCKET_NETWORK socket1170 6549 driver21b
SOCKET_NETWORK socket1171 6550 driver21c
SOCKET_NETWORK socket1172 6551 driver21d
SOCKET_NETWORK socket1173 6552 driver21e
SOCKET_NETWORK socket1174 6553 driver21f
SOCKET_NETWORK socket1175 6554 driver21a
SOCKET_NETWORK socket1176 6555 driver21b
SOCKET_NETWORK socket1177 6556 driver22a
SOCKET_NETWORK socket1178 6557 driver22b
SOCKET_NETWORK socket1179 6558 driver22c
SOCKET_NETWORK socket1180 6559 driver22d
SOCKET_NETWORK socket1181 6560 driver22e
SOCKET_NETWORK socket1182 6561 driver22f
SOCKET_NETWORK socket1183 6562 driver22a
SOCKET_NETWORK socket1184 6563 driver22b
SOCKET_NETWORK socket1185 6564 driver22c
SOCKET_NETWORK socket1186 6565 driver22d
SOCKET_NETWORK socket1187 6566 driver22e
SOCKET_NETWORK socket1188 6567 driver22f
SOCKET_NETWORK socket1189 6568 driver22a
SOCKET_NETWORK socket1190 6569 driver22b
SOCKET_NETWORK socket1191 6570 driver22c
SOCKET_NETWORK socket1192 6571 driver22d
SOCKET_NETWORK socket1193 6572 driver22e
SOCKET_NETWORK socket1194 6573 driver22f
SOCKET_NETWORK socket1195 6574 driver22a
SOCKET_NETWORK socket1196 6575 driver22b
SOCKET_NETWORK socket1197 6576 driver22c
SOCKET_NETWORK socket1198 6577 driver22d
SOCKET_NETWORK socket1199 6578 driver22e
SOCKET_NETWORK socket1200 6579 driver22f
SOCKET_NETWORK socket1201 6580 driver22a
SOCKET_NETWORK socket1202 6581 driver22b
SOCKET_NETWORK socket1203 6582 driver22c
SOCKET_NETWORK socket1204 6583 driver22d
SOCKET_NETWORK socket1205 6584 driver22e
SOCKET_NETWORK socket1206 6585 driver22f
SOCKET_NETWORK socket1207 6586 driver22a
SOCKET_NETWORK socket1208 6587 driver22b
SOCKET_NETWORK socket1209 6588 driver22c
SOCKET_NETWORK socket1210 6589 driver22d
SOCKET_NETWORK socket1211 6590 driver22e
SOCKET_NETWORK socket1212 6591 driver22f
SOCKET_NETWORK socket1213 6592 driver22a
SOCKET_NETWORK socket1214 6593 driver22b
SOCKET_NETWORK socket1215 6594 driver22c
SOCKET_NETWORK socket1216 6595 driver22d
SOCKET_NETWORK socket1217 6596 driver22e
SOCKET_NETWORK socket1218 6597 driver22f
SOCKET_NETWORK socket1219 6598 driver22a
SOCKET_NETWORK socket1220 6599 driver22b
SOCKET_NETWORK socket1221 6600 driver22c
SOCKET_NETWORK socket1222 6601 driver22d
SOCKET_NETWORK socket1223 6602 driver22e
SOCKET_NETWORK socket1224 6603 driver22f
SOCKET_NETWORK socket1225 6604 driver22a
SOCKET_NETWORK socket1226 6605 driver22b
SOCKET_NETWORK socket1227 6606 driver22c
SOCKET_NETWORK socket1228 6607 driver22d
SOCKET_NETWORK socket1229 6608 driver22e
SOCKET_NETWORK socket1230 6609 driver22f
SOCKET_NETWORK socket1231 6610 driver22a
SOCKET_NETWORK socket1232 6611 driver22b
SOCKET_NETWORK socket1233 6500 driver23a
SOCKET_NETWORK socket1234 6501 driver23b
SOCKET_NETWORK socket1235 6502 driver23c
SOCKET_NETWORK socket1236 6503 driver23d
SOCKET_NETWORK socket1237 6504 driver23e
SOCKET_NETWORK socket1238 6505 driver23f
SOCKET_NETWORK socket1239 6506 driver23a
SOCKET_NETWORK socket1240 6507 driver23b
SOCKET_NETWORK socket1241 6508 driver23c
SOCKET_NETWORK socket1242 6509 driver23d
SOCKET_NETWORK socket1243 6510 driver23e
SOCKET_NETWORK socket1244 6511 driver23f
SOCKET_NETWORK socket1245 6512 driver23a

```
SOCKET_NETWORK socket1246  6513 driver23b
SOCKET_NETWORK socket1247  6514 driver23c
SOCKET_NETWORK socket1248  6515 driver23d
SOCKET_NETWORK socket1249  6516 driver23e
SOCKET_NETWORK socket1250  6517 driver23f
SOCKET_NETWORK socket1251  6518 driver23a
SOCKET_NETWORK socket1252  6519 driver23b
SOCKET_NETWORK socket1253  6520 driver23c
SOCKET_NETWORK socket1254  6521 driver23d
SOCKET_NETWORK socket1255  6522 driver23e
SOCKET_NETWORK socket1256  6523 driver23f
SOCKET_NETWORK socket1257  6524 driver23a
SOCKET_NETWORK socket1258  6525 driver23b
SOCKET_NETWORK socket1259  6526 driver23c
SOCKET_NETWORK socket1260  6527 driver23d
SOCKET_NETWORK socket1261  6528 driver23e
SOCKET_NETWORK socket1262  6529 driver23f
SOCKET_NETWORK socket1263  6530 driver23a
SOCKET_NETWORK socket1264  6531 driver23b
SOCKET_NETWORK socket1265  6532 driver23c
SOCKET_NETWORK socket1266  6533 driver23d
SOCKET_NETWORK socket1267  6534 driver23e
SOCKET_NETWORK socket1268  6535 driver23f
SOCKET_NETWORK socket1269  6536 driver23a
SOCKET_NETWORK socket1270  6537 driver23b
SOCKET_NETWORK socket1271  6538 driver23c
SOCKET_NETWORK socket1272  6539 driver23d
SOCKET_NETWORK socket1273  6540 driver23e
SOCKET_NETWORK socket1274  6541 driver23f
SOCKET_NETWORK socket1275  6542 driver23a
SOCKET_NETWORK socket1276  6543 driver23b
SOCKET_NETWORK socket1277  6544 driver23c
SOCKET_NETWORK socket1278  6545 driver23d
SOCKET_NETWORK socket1279  6546 driver23e
SOCKET_NETWORK socket1280  6547 driver23f
SOCKET_NETWORK socket1281  6548 driver23a
SOCKET_NETWORK socket1282  6549 driver23b
SOCKET_NETWORK socket1283  6550 driver23c
SOCKET_NETWORK socket1284  6551 driver23d
SOCKET_NETWORK socket1285  6552 driver23e
SOCKET_NETWORK socket1286  6553 driver23f
SOCKET_NETWORK socket1287  6554 driver23a
SOCKET_NETWORK socket1288  6555 driver23b
SOCKET_NETWORK socket1289  6556 driver24a
SOCKET_NETWORK socket1290  6557 driver24b
SOCKET_NETWORK socket1291  6558 driver24c
SOCKET_NETWORK socket1292  6559 driver24d
SOCKET_NETWORK socket1293  6560 driver24e
SOCKET_NETWORK socket1294  6561 driver24f
SOCKET_NETWORK socket1295  6562 driver24a
SOCKET_NETWORK socket1296  6563 driver24b
SOCKET_NETWORK socket1297  6564 driver24c
SOCKET_NETWORK socket1298  6565 driver24d
SOCKET_NETWORK socket1299  6566 driver24e
SOCKET_NETWORK socket1300  6567 driver24f
SOCKET_NETWORK socket1301  6568 driver24a
SOCKET_NETWORK socket1302  6569 driver24b
SOCKET_NETWORK socket1303  6570 driver24c
SOCKET_NETWORK socket1304  6571 driver24d
SOCKET_NETWORK socket1305  6572 driver24e
SOCKET_NETWORK socket1306  6573 driver24f
SOCKET_NETWORK socket1307  6574 driver24a
SOCKET_NETWORK socket1308  6575 driver24b
SOCKET_NETWORK socket1309  6576 driver24c
SOCKET_NETWORK socket1310  6577 driver24d
SOCKET_NETWORK socket1311  6578 driver24e
SOCKET_NETWORK socket1312  6579 driver24f
SOCKET_NETWORK socket1313  6580 driver24a
SOCKET_NETWORK socket1314  6581 driver24b
SOCKET_NETWORK socket1315  6582 driver24c
SOCKET_NETWORK socket1316  6583 driver24d
SOCKET_NETWORK socket1317  6584 driver24e
SOCKET_NETWORK socket1318  6585 driver24f
SOCKET_NETWORK socket1319  6586 driver24a
SOCKET_NETWORK socket1320  6587 driver24b
SOCKET_NETWORK socket1321  6588 driver24c
SOCKET_NETWORK socket1322  6589 driver24d
SOCKET_NETWORK socket1323  6590 driver24e
SOCKET_NETWORK socket1324  6591 driver24f
SOCKET_NETWORK socket1325  6592 driver24a
SOCKET_NETWORK socket1326  6593 driver24b
SOCKET_NETWORK socket1327  6594 driver24c
SOCKET_NETWORK socket1328  6595 driver24d
SOCKET_NETWORK socket1329  6596 driver24e
SOCKET_NETWORK socket1330  6597 driver24f
SOCKET_NETWORK socket1331  6598 driver24a
SOCKET_NETWORK socket1332  6599 driver24b
SOCKET_NETWORK socket1333  6600 driver24c
SOCKET_NETWORK socket1334  6601 driver24d
SOCKET_NETWORK socket1335  6602 driver24e
SOCKET_NETWORK socket1336  6603 driver24f
SOCKET_NETWORK socket1337  6604 driver24a
SOCKET_NETWORK socket1338  6605 driver24b
SOCKET_NETWORK socket1339  6606 driver24c
SOCKET_NETWORK socket1340  6607 driver24d
SOCKET_NETWORK socket1341  6608 driver24e
SOCKET_NETWORK socket1342  6609 driver24f
SOCKET_NETWORK socket1343  6610 driver24a
SOCKET_NETWORK socket1344  6611 driver24b
```

```
#elif MASTER_NUM5
SOCKET_NETWORK socket1345  6500 driver25a
SOCKET_NETWORK socket1346  6501 driver25b
SOCKET_NETWORK socket1347  6502 driver25c
SOCKET_NETWORK socket1348  6503 driver25d
SOCKET_NETWORK socket1349  6504 driver25e
SOCKET_NETWORK socket1350  6505 driver25f
SOCKET_NETWORK socket1351  6506 driver25a
SOCKET_NETWORK socket1352  6507 driver25b
SOCKET_NETWORK socket1353  6508 driver25c
SOCKET_NETWORK socket1354  6509 driver25d
SOCKET_NETWORK socket1355  6510 driver25e
SOCKET_NETWORK socket1356  6511 driver25f
SOCKET_NETWORK socket1357  6512 driver25a
SOCKET_NETWORK socket1358  6513 driver25b
SOCKET_NETWORK socket1359  6514 driver25c
SOCKET_NETWORK socket1360  6515 driver25d
SOCKET_NETWORK socket1361  6516 driver25e
SOCKET_NETWORK socket1362  6517 driver25f
SOCKET_NETWORK socket1363  6518 driver25a
SOCKET_NETWORK socket1364  6519 driver25b
SOCKET_NETWORK socket1365  6520 driver25c
SOCKET_NETWORK socket1366  6521 driver25d
SOCKET_NETWORK socket1367  6522 driver25e
SOCKET_NETWORK socket1368  6523 driver25f
SOCKET_NETWORK socket1369  6524 driver25a
SOCKET_NETWORK socket1370  6525 driver25b
SOCKET_NETWORK socket1371  6526 driver25c
SOCKET_NETWORK socket1372  6527 driver25d
SOCKET_NETWORK socket1373  6528 driver25e
SOCKET_NETWORK socket1374  6529 driver25f
SOCKET_NETWORK socket1375  6530 driver25a
SOCKET_NETWORK socket1376  6531 driver25b
SOCKET_NETWORK socket1377  6532 driver25c
SOCKET_NETWORK socket1378  6533 driver25d
SOCKET_NETWORK socket1379  6534 driver25e
SOCKET_NETWORK socket1380  6535 driver25f
SOCKET_NETWORK socket1381  6536 driver25a
SOCKET_NETWORK socket1382  6537 driver25b
SOCKET_NETWORK socket1383  6538 driver25c
SOCKET_NETWORK socket1384  6539 driver25d
SOCKET_NETWORK socket1385  6540 driver25e
SOCKET_NETWORK socket1386  6541 driver25f
SOCKET_NETWORK socket1387  6542 driver25a
SOCKET_NETWORK socket1388  6543 driver25b
SOCKET_NETWORK socket1389  6544 driver25c
SOCKET_NETWORK socket1390  6545 driver25d
SOCKET_NETWORK socket1391  6546 driver25e
SOCKET_NETWORK socket1392  6547 driver25f
SOCKET_NETWORK socket1393  6548 driver25a
SOCKET_NETWORK socket1394  6549 driver25b
SOCKET_NETWORK socket1395  6550 driver25c
SOCKET_NETWORK socket1396  6551 driver25d
SOCKET_NETWORK socket1397  6552 driver25e
SOCKET_NETWORK socket1398  6553 driver25f
SOCKET_NETWORK socket1399  6554 driver25a
SOCKET_NETWORK socket1400  6555 driver25b
SOCKET_NETWORK socket1401  6556 driver26a
SOCKET_NETWORK socket1402  6557 driver26b
SOCKET_NETWORK socket1403  6558 driver26c
SOCKET_NETWORK socket1404  6559 driver26d
SOCKET_NETWORK socket1405  6560 driver26e
SOCKET_NETWORK socket1406  6561 driver26f
SOCKET_NETWORK socket1407  6562 driver26a
SOCKET_NETWORK socket1408  6563 driver26b
SOCKET_NETWORK socket1409  6564 driver26c
SOCKET_NETWORK socket1410  6565 driver26d
SOCKET_NETWORK socket1411  6566 driver26e
SOCKET_NETWORK socket1412  6567 driver26f
SOCKET_NETWORK socket1413  6568 driver26a
SOCKET_NETWORK socket1414  6569 driver26b
SOCKET_NETWORK socket1415  6570 driver26c
SOCKET_NETWORK socket1416  6571 driver26d
SOCKET_NETWORK socket1417  6572 driver26e
SOCKET_NETWORK socket1418  6573 driver26f
SOCKET_NETWORK socket1419  6574 driver26a
SOCKET_NETWORK socket1420  6575 driver26b
SOCKET_NETWORK socket1421  6576 driver26c
SOCKET_NETWORK socket1422  6577 driver26d
SOCKET_NETWORK socket1423  6578 driver26e
SOCKET_NETWORK socket1424  6579 driver26f
SOCKET_NETWORK socket1425  6580 driver26a
SOCKET_NETWORK socket1426  6581 driver26b
SOCKET_NETWORK socket1427  6582 driver26c
SOCKET_NETWORK socket1428  6583 driver26d
SOCKET_NETWORK socket1429  6584 driver26e
SOCKET_NETWORK socket1430  6585 driver26f
SOCKET_NETWORK socket1431  6586 driver26a
SOCKET_NETWORK socket1432  6587 driver26b
SOCKET_NETWORK socket1433  6588 driver26c
SOCKET_NETWORK socket1434  6589 driver26d
SOCKET_NETWORK socket1435  6590 driver26e
SOCKET_NETWORK socket1436  6591 driver26f
SOCKET_NETWORK socket1437  6592 driver26a
SOCKET_NETWORK socket1438  6593 driver26b
SOCKET_NETWORK socket1439  6594 driver26c
SOCKET_NETWORK socket1440  6595 driver26d
SOCKET_NETWORK socket1441  6596 driver26e
SOCKET_NETWORK socket1442  6597 driver26f
```

SOCKET_NETWORK socket1443 6598 driver26a
SOCKET_NETWORK socket1444 6599 driver26b
SOCKET_NETWORK socket1445 6600 driver26c
SOCKET_NETWORK socket1446 6601 driver26d
SOCKET_NETWORK socket1447 6602 driver26e
SOCKET_NETWORK socket1448 6603 driver26f
SOCKET_NETWORK socket1449 6604 driver26a
SOCKET_NETWORK socket1450 6605 driver26b
SOCKET_NETWORK socket1451 6606 driver26c
SOCKET_NETWORK socket1452 6607 driver26d
SOCKET_NETWORK socket1453 6608 driver26e
SOCKET_NETWORK socket1454 6609 driver26f
SOCKET_NETWORK socket1455 6610 driver26a
SOCKET_NETWORK socket1456 6611 driver26b
SOCKET_NETWORK socket1457 6500 driver27a
SOCKET_NETWORK socket1458 6501 driver27b
SOCKET_NETWORK socket1459 6502 driver27c
SOCKET_NETWORK socket1460 6503 driver27d
SOCKET_NETWORK socket1461 6504 driver27e
SOCKET_NETWORK socket1462 6505 driver27f
SOCKET_NETWORK socket1463 6506 driver27a
SOCKET_NETWORK socket1464 6507 driver27b
SOCKET_NETWORK socket1465 6508 driver27c
SOCKET_NETWORK socket1466 6509 driver27d
SOCKET_NETWORK socket1467 6510 driver27e
SOCKET_NETWORK socket1468 6511 driver27f
SOCKET_NETWORK socket1469 6512 driver27a
SOCKET_NETWORK socket1470 6513 driver27b
SOCKET_NETWORK socket1471 6514 driver27c
SOCKET_NETWORK socket1472 6515 driver27d
SOCKET_NETWORK socket1473 6516 driver27e
SOCKET_NETWORK socket1474 6517 driver27f
SOCKET_NETWORK socket1475 6518 driver27a
SOCKET_NETWORK socket1476 6519 driver27b
SOCKET_NETWORK socket1477 6520 driver27c
SOCKET_NETWORK socket1478 6521 driver27d
SOCKET_NETWORK socket1479 6522 driver27e
SOCKET_NETWORK socket1480 6523 driver27f
SOCKET_NETWORK socket1481 6524 driver27a
SOCKET_NETWORK socket1482 6525 driver27b
SOCKET_NETWORK socket1483 6526 driver27c
SOCKET_NETWORK socket1484 6527 driver27d
SOCKET_NETWORK socket1485 6528 driver27e
SOCKET_NETWORK socket1486 6529 driver27f
SOCKET_NETWORK socket1487 6530 driver27a
SOCKET_NETWORK socket1488 6531 driver27b
SOCKET_NETWORK socket1489 6532 driver27c
SOCKET_NETWORK socket1490 6533 driver27d
SOCKET_NETWORK socket1491 6534 driver27e
SOCKET_NETWORK socket1492 6535 driver27f
SOCKET_NETWORK socket1493 6536 driver27a
SOCKET_NETWORK socket1494 6537 driver27b
SOCKET_NETWORK socket1495 6538 driver27c
SOCKET_NETWORK socket1496 6539 driver27d
SOCKET_NETWORK socket1497 6540 driver27e
SOCKET_NETWORK socket1498 6541 driver27f
SOCKET_NETWORK socket1499 6542 driver27a
SOCKET_NETWORK socket1500 6543 driver27b
SOCKET_NETWORK socket1501 6544 driver27c
SOCKET_NETWORK socket1502 6545 driver27d
SOCKET_NETWORK socket1503 6546 driver27e
SOCKET_NETWORK socket1504 6547 driver27f
SOCKET_NETWORK socket1505 6548 driver27a
SOCKET_NETWORK socket1506 6549 driver27b
SOCKET_NETWORK socket1507 6550 driver27c
SOCKET_NETWORK socket1508 6551 driver27d
SOCKET_NETWORK socket1509 6552 driver27e
SOCKET_NETWORK socket1510 6553 driver27f
SOCKET_NETWORK socket1511 6554 driver27a
SOCKET_NETWORK socket1512 6555 driver27b
SOCKET_NETWORK socket1513 6556 driver28a
SOCKET_NETWORK socket1514 6557 driver28b
SOCKET_NETWORK socket1515 6558 driver28c
SOCKET_NETWORK socket1516 6559 driver28d
SOCKET_NETWORK socket1517 6560 driver28e
SOCKET_NETWORK socket1518 6561 driver28f
SOCKET_NETWORK socket1519 6562 driver28a
SOCKET_NETWORK socket1520 6563 driver28b
SOCKET_NETWORK socket1521 6564 driver28c
SOCKET_NETWORK socket1522 6565 driver28d
SOCKET_NETWORK socket1523 6566 driver28e
SOCKET_NETWORK socket1524 6567 driver28f
SOCKET_NETWORK socket1525 6568 driver28a
SOCKET_NETWORK socket1526 6569 driver28b
SOCKET_NETWORK socket1527 6570 driver28c
SOCKET_NETWORK socket1528 6571 driver28d
SOCKET_NETWORK socket1529 6572 driver28e
SOCKET_NETWORK socket1530 6573 driver28f
SOCKET_NETWORK socket1531 6574 driver28a
SOCKET_NETWORK socket1532 6575 driver28b
SOCKET_NETWORK socket1533 6576 driver28c
SOCKET_NETWORK socket1534 6577 driver28d
SOCKET_NETWORK socket1535 6578 driver28e
SOCKET_NETWORK socket1536 6579 driver28f
SOCKET_NETWORK socket1537 6580 driver28a
SOCKET_NETWORK socket1538 6581 driver28b
SOCKET_NETWORK socket1539 6582 driver28c
SOCKET_NETWORK socket1540 6583 driver28d
SOCKET_NETWORK socket1541 6584 driver28e
SOCKET_NETWORK socket1542 6585 driver28f
SOCKET_NETWORK socket1543 6586 driver28a
SOCKET_NETWORK socket1544 6587 driver28b
SOCKET_NETWORK socket1545 6588 driver28c
SOCKET_NETWORK socket1546 6589 driver28d
SOCKET_NETWORK socket1547 6590 driver28e
SOCKET_NETWORK socket1548 6591 driver28f
SOCKET_NETWORK socket1549 6592 driver28a
SOCKET_NETWORK socket1550 6593 driver28b
SOCKET_NETWORK socket1551 6594 driver28c
SOCKET_NETWORK socket1552 6595 driver28d
SOCKET_NETWORK socket1553 6596 driver28e
SOCKET_NETWORK socket1554 6597 driver28f
SOCKET_NETWORK socket1555 6598 driver28a
SOCKET_NETWORK socket1556 6599 driver28b
SOCKET_NETWORK socket1557 6600 driver28c
SOCKET_NETWORK socket1558 6601 driver28d
SOCKET_NETWORK socket1559 6602 driver28e
SOCKET_NETWORK socket1560 6603 driver28f
SOCKET_NETWORK socket1561 6604 driver28a
SOCKET_NETWORK socket1562 6605 driver28b
SOCKET_NETWORK socket1563 6606 driver28c
SOCKET_NETWORK socket1564 6607 driver28d
SOCKET_NETWORK socket1565 6608 driver28e
SOCKET_NETWORK socket1566 6609 driver28f
SOCKET_NETWORK socket1567 6610 driver28a
SOCKET_NETWORK socket1568 6611 driver28b
SOCKET_NETWORK socket1569 6500 driver29a
SOCKET_NETWORK socket1570 6501 driver29b
SOCKET_NETWORK socket1571 6502 driver29c
SOCKET_NETWORK socket1572 6503 driver29d
SOCKET_NETWORK socket1573 6504 driver29e
SOCKET_NETWORK socket1574 6505 driver29f
SOCKET_NETWORK socket1575 6506 driver29a
SOCKET_NETWORK socket1576 6507 driver29b
SOCKET_NETWORK socket1577 6508 driver29c
SOCKET_NETWORK socket1578 6509 driver29d
SOCKET_NETWORK socket1579 6510 driver29e
SOCKET_NETWORK socket1580 6511 driver29f
SOCKET_NETWORK socket1581 6512 driver29a
SOCKET_NETWORK socket1582 6513 driver29b
SOCKET_NETWORK socket1583 6514 driver29c
SOCKET_NETWORK socket1584 6515 driver29d
SOCKET_NETWORK socket1585 6516 driver29e
SOCKET_NETWORK socket1586 6517 driver29f
SOCKET_NETWORK socket1587 6518 driver29a
SOCKET_NETWORK socket1588 6519 driver29b
SOCKET_NETWORK socket1589 6520 driver29c
SOCKET_NETWORK socket1590 6521 driver29d
SOCKET_NETWORK socket1591 6522 driver29e
SOCKET_NETWORK socket1592 6523 driver29f
SOCKET_NETWORK socket1593 6524 driver29a
SOCKET_NETWORK socket1594 6525 driver29b
SOCKET_NETWORK socket1595 6526 driver29c
SOCKET_NETWORK socket1596 6527 driver29d
SOCKET_NETWORK socket1597 6528 driver29e
SOCKET_NETWORK socket1598 6529 driver29f
SOCKET_NETWORK socket1599 6530 driver29a
SOCKET_NETWORK socket1600 6531 driver29b
SOCKET_NETWORK socket1601 6532 driver29c
SOCKET_NETWORK socket1602 6533 driver29d
SOCKET_NETWORK socket1603 6534 driver29e
SOCKET_NETWORK socket1604 6535 driver29f
SOCKET_NETWORK socket1605 6536 driver29a
SOCKET_NETWORK socket1606 6537 driver29b
SOCKET_NETWORK socket1607 6538 driver29c
SOCKET_NETWORK socket1608 6539 driver29d
SOCKET_NETWORK socket1609 6540 driver29e
SOCKET_NETWORK socket1610 6541 driver29f
SOCKET_NETWORK socket1611 6542 driver29a
SOCKET_NETWORK socket1612 6543 driver29b
SOCKET_NETWORK socket1613 6544 driver29c
SOCKET_NETWORK socket1614 6545 driver29d
SOCKET_NETWORK socket1615 6546 driver29e
SOCKET_NETWORK socket1616 6547 driver29f
SOCKET_NETWORK socket1617 6548 driver29a
SOCKET_NETWORK socket1618 6549 driver29b
SOCKET_NETWORK socket1619 6550 driver29c
SOCKET_NETWORK socket1620 6551 driver29d
SOCKET_NETWORK socket1621 6552 driver29e
SOCKET_NETWORK socket1622 6553 driver29f
SOCKET_NETWORK socket1623 6554 driver29a
SOCKET_NETWORK socket1624 6555 driver29b
SOCKET_NETWORK socket1625 6556 driver30a
SOCKET_NETWORK socket1626 6557 driver30b
SOCKET_NETWORK socket1627 6558 driver30c
SOCKET_NETWORK socket1628 6559 driver30d
SOCKET_NETWORK socket1629 6560 driver30e
SOCKET_NETWORK socket1630 6561 driver30f
SOCKET_NETWORK socket1631 6562 driver30a
SOCKET_NETWORK socket1632 6563 driver30b
SOCKET_NETWORK socket1633 6564 driver30c
SOCKET_NETWORK socket1634 6565 driver30d
SOCKET_NETWORK socket1635 6566 driver30e
SOCKET_NETWORK socket1636 6567 driver30f
SOCKET_NETWORK socket1637 6568 driver30a
SOCKET_NETWORK socket1638 6569 driver30b
SOCKET_NETWORK socket1639 6570 driver30c
SOCKET_NETWORK socket1640 6571 driver30d

```
SOCKET_NETWORK socket1641  6572 driver30e
SOCKET_NETWORK socket1642  6573 driver30f
SOCKET_NETWORK socket1643  6574 driver30a
SOCKET_NETWORK socket1644  6575 driver30b
SOCKET_NETWORK socket1645  6576 driver30c
SOCKET_NETWORK socket1646  6577 driver30d
SOCKET_NETWORK socket1647  6578 driver30e
SOCKET_NETWORK socket1648  6579 driver30f
SOCKET_NETWORK socket1649  6580 driver30a
SOCKET_NETWORK socket1650  6581 driver30b
SOCKET_NETWORK socket1651  6582 driver30c
SOCKET_NETWORK socket1652  6583 driver30d
SOCKET_NETWORK socket1653  6584 driver30e
SOCKET_NETWORK socket1654  6585 driver30f
SOCKET_NETWORK socket1655  6586 driver30a
SOCKET_NETWORK socket1656  6587 driver30b
SOCKET_NETWORK socket1657  6588 driver30c
SOCKET_NETWORK socket1658  6589 driver30d
SOCKET_NETWORK socket1659  6590 driver30e
SOCKET_NETWORK socket1660  6591 driver30f
SOCKET_NETWORK socket1661  6592 driver30a
SOCKET_NETWORK socket1662  6593 driver30b
SOCKET_NETWORK socket1663  6594 driver30c
SOCKET_NETWORK socket1664  6595 driver30d
SOCKET_NETWORK socket1665  6596 driver30e
SOCKET_NETWORK socket1666  6597 driver30f
SOCKET_NETWORK socket1667  6598 driver30a
SOCKET_NETWORK socket1668  6599 driver30b
SOCKET_NETWORK socket1669  6600 driver30c
SOCKET_NETWORK socket1670  6601 driver30d
SOCKET_NETWORK socket1671  6602 driver30e
SOCKET_NETWORK socket1672  6603 driver30f
SOCKET_NETWORK socket1673  6604 driver30a
SOCKET_NETWORK socket1674  6605 driver30b
SOCKET_NETWORK socket1675  6606 driver30c
SOCKET_NETWORK socket1676  6607 driver30d
SOCKET_NETWORK socket1677  6608 driver30e
SOCKET_NETWORK socket1678  6609 driver30f
SOCKET_NETWORK socket1679  6610 driver30a
SOCKET_NETWORK socket1680  6611 driver30b
#elif MASTER_NUM6
SOCKET_NETWORK socket1681  6500 driver31a
SOCKET_NETWORK socket1682  6501 driver31b
SOCKET_NETWORK socket1683  6502 driver31c
SOCKET_NETWORK socket1684  6503 driver31d
SOCKET_NETWORK socket1685  6504 driver31e
SOCKET_NETWORK socket1686  6505 driver31f
SOCKET_NETWORK socket1687  6506 driver31a
SOCKET_NETWORK socket1688  6507 driver31b
SOCKET_NETWORK socket1689  6508 driver31c
SOCKET_NETWORK socket1690  6509 driver31d
SOCKET_NETWORK socket1691  6510 driver31e
SOCKET_NETWORK socket1692  6511 driver31f
SOCKET_NETWORK socket1693  6512 driver31a
SOCKET_NETWORK socket1694  6513 driver31b
SOCKET_NETWORK socket1695  6514 driver31c
SOCKET_NETWORK socket1696  6515 driver31d
SOCKET_NETWORK socket1697  6516 driver31e
SOCKET_NETWORK socket1698  6517 driver31f
SOCKET_NETWORK socket1699  6518 driver31a
SOCKET_NETWORK socket1700  6519 driver31b
SOCKET_NETWORK socket1701  6520 driver31c
SOCKET_NETWORK socket1702  6521 driver31d
SOCKET_NETWORK socket1703  6522 driver31e
SOCKET_NETWORK socket1704  6523 driver31f
SOCKET_NETWORK socket1705  6524 driver31a
SOCKET_NETWORK socket1706  6525 driver31b
SOCKET_NETWORK socket1707  6526 driver31c
SOCKET_NETWORK socket1708  6527 driver31d
SOCKET_NETWORK socket1709  6528 driver31e
SOCKET_NETWORK socket1710  6529 driver31f
SOCKET_NETWORK socket1711  6530 driver31a
SOCKET_NETWORK socket1712  6531 driver31b
SOCKET_NETWORK socket1713  6532 driver31c
SOCKET_NETWORK socket1714  6533 driver31d
SOCKET_NETWORK socket1715  6534 driver31e
SOCKET_NETWORK socket1716  6535 driver31f
SOCKET_NETWORK socket1717  6536 driver31a
SOCKET_NETWORK socket1718  6537 driver31b
SOCKET_NETWORK socket1719  6538 driver31c
SOCKET_NETWORK socket1720  6539 driver31d
SOCKET_NETWORK socket1721  6540 driver31e
SOCKET_NETWORK socket1722  6541 driver31f
SOCKET_NETWORK socket1723  6542 driver31a
SOCKET_NETWORK socket1724  6543 driver31b
SOCKET_NETWORK socket1725  6544 driver31c
SOCKET_NETWORK socket1726  6545 driver31d
SOCKET_NETWORK socket1727  6546 driver31e
SOCKET_NETWORK socket1728  6547 driver31f
SOCKET_NETWORK socket1729  6548 driver31a
SOCKET_NETWORK socket1730  6549 driver31b
SOCKET_NETWORK socket1731  6550 driver31c
SOCKET_NETWORK socket1732  6551 driver31d
SOCKET_NETWORK socket1733  6552 driver31e
SOCKET_NETWORK socket1734  6553 driver31f
SOCKET_NETWORK socket1735  6554 driver31a
SOCKET_NETWORK socket1736  6555 driver31b
SOCKET_NETWORK socket1737  6556 driver32a
SOCKET_NETWORK socket1738  6557 driver32b
SOCKET_NETWORK socket1739  6558 driver32c
SOCKET_NETWORK socket1740  6559 driver32d
SOCKET_NETWORK socket1741  6560 driver32e
SOCKET_NETWORK socket1742  6561 driver32f
SOCKET_NETWORK socket1743  6562 driver32a
SOCKET_NETWORK socket1744  6563 driver32b
SOCKET_NETWORK socket1745  6564 driver32c
SOCKET_NETWORK socket1746  6565 driver32d
SOCKET_NETWORK socket1747  6566 driver32e
SOCKET_NETWORK socket1748  6567 driver32f
SOCKET_NETWORK socket1749  6568 driver32a
SOCKET_NETWORK socket1750  6569 driver32b
SOCKET_NETWORK socket1751  6570 driver32c
SOCKET_NETWORK socket1752  6571 driver32d
SOCKET_NETWORK socket1753  6572 driver32e
SOCKET_NETWORK socket1754  6573 driver32f
SOCKET_NETWORK socket1755  6574 driver32a
SOCKET_NETWORK socket1756  6575 driver32b
SOCKET_NETWORK socket1757  6576 driver32c
SOCKET_NETWORK socket1758  6577 driver32d
SOCKET_NETWORK socket1759  6578 driver32e
SOCKET_NETWORK socket1760  6579 driver32f
SOCKET_NETWORK socket1761  6580 driver32a
SOCKET_NETWORK socket1762  6581 driver32b
SOCKET_NETWORK socket1763  6582 driver32c
SOCKET_NETWORK socket1764  6583 driver32d
SOCKET_NETWORK socket1765  6584 driver32e
SOCKET_NETWORK socket1766  6585 driver32f
SOCKET_NETWORK socket1767  6586 driver32a
SOCKET_NETWORK socket1768  6587 driver32b
SOCKET_NETWORK socket1769  6588 driver32c
SOCKET_NETWORK socket1770  6589 driver32d
SOCKET_NETWORK socket1771  6590 driver32e
SOCKET_NETWORK socket1772  6591 driver32f
SOCKET_NETWORK socket1773  6592 driver32a
SOCKET_NETWORK socket1774  6593 driver32b
SOCKET_NETWORK socket1775  6594 driver32c
SOCKET_NETWORK socket1776  6595 driver32d
SOCKET_NETWORK socket1777  6596 driver32e
SOCKET_NETWORK socket1778  6597 driver32f
SOCKET_NETWORK socket1779  6598 driver32a
SOCKET_NETWORK socket1780  6599 driver32b
SOCKET_NETWORK socket1781  6600 driver32c
SOCKET_NETWORK socket1782  6601 driver32d
SOCKET_NETWORK socket1783  6602 driver32e
SOCKET_NETWORK socket1784  6603 driver32f
SOCKET_NETWORK socket1785  6604 driver32a
SOCKET_NETWORK socket1786  6605 driver32b
SOCKET_NETWORK socket1787  6606 driver32c
SOCKET_NETWORK socket1788  6607 driver32d
SOCKET_NETWORK socket1789  6608 driver32e
SOCKET_NETWORK socket1790  6609 driver32f
SOCKET_NETWORK socket1791  6610 driver32a
SOCKET_NETWORK socket1792  6611 driver32b
SOCKET_NETWORK socket1793  6500 driver33a
SOCKET_NETWORK socket1794  6501 driver33b
SOCKET_NETWORK socket1795  6502 driver33c
SOCKET_NETWORK socket1796  6503 driver33d
SOCKET_NETWORK socket1797  6504 driver33e
SOCKET_NETWORK socket1798  6505 driver33f
SOCKET_NETWORK socket1799  6506 driver33a
SOCKET_NETWORK socket1800  6507 driver33b
SOCKET_NETWORK socket1801  6508 driver33c
SOCKET_NETWORK socket1802  6509 driver33d
SOCKET_NETWORK socket1803  6510 driver33e
SOCKET_NETWORK socket1804  6511 driver33f
SOCKET_NETWORK socket1805  6512 driver33a
SOCKET_NETWORK socket1806  6513 driver33b
SOCKET_NETWORK socket1807  6514 driver33c
SOCKET_NETWORK socket1808  6515 driver33d
SOCKET_NETWORK socket1809  6516 driver33e
SOCKET_NETWORK socket1810  6517 driver33f
SOCKET_NETWORK socket1811  6518 driver33a
SOCKET_NETWORK socket1812  6519 driver33b
SOCKET_NETWORK socket1813  6520 driver33c
SOCKET_NETWORK socket1814  6521 driver33d
SOCKET_NETWORK socket1815  6522 driver33e
SOCKET_NETWORK socket1816  6523 driver33f
SOCKET_NETWORK socket1817  6524 driver33a
SOCKET_NETWORK socket1818  6525 driver33b
SOCKET_NETWORK socket1819  6526 driver33c
SOCKET_NETWORK socket1820  6527 driver33d
SOCKET_NETWORK socket1821  6528 driver33e
SOCKET_NETWORK socket1822  6529 driver33f
SOCKET_NETWORK socket1823  6530 driver33a
SOCKET_NETWORK socket1824  6531 driver33b
SOCKET_NETWORK socket1825  6532 driver33c
SOCKET_NETWORK socket1826  6533 driver33d
SOCKET_NETWORK socket1827  6534 driver33e
SOCKET_NETWORK socket1828  6535 driver33f
SOCKET_NETWORK socket1829  6536 driver33a
SOCKET_NETWORK socket1830  6537 driver33b
SOCKET_NETWORK socket1831  6538 driver33c
SOCKET_NETWORK socket1832  6539 driver33d
SOCKET_NETWORK socket1833  6540 driver33e
SOCKET_NETWORK socket1834  6541 driver33f
SOCKET_NETWORK socket1835  6542 driver33a
SOCKET_NETWORK socket1836  6543 driver33b
SOCKET_NETWORK socket1837  6544 driver33c
```

SOCKET_NETWORK socket1838 6545 driver33d
SOCKET_NETWORK socket1839 6546 driver33e
SOCKET_NETWORK socket1840 6547 driver33f
SOCKET_NETWORK socket1841 6548 driver33a
SOCKET_NETWORK socket1842 6549 driver33b
SOCKET_NETWORK socket1843 6550 driver33c
SOCKET_NETWORK socket1844 6551 driver33d
SOCKET_NETWORK socket1845 6552 driver33e
SOCKET_NETWORK socket1846 6553 driver33f
SOCKET_NETWORK socket1847 6554 driver33a
SOCKET_NETWORK socket1848 6555 driver33b
SOCKET_NETWORK socket1849 6556 driver34a
SOCKET_NETWORK socket1850 6557 driver34b
SOCKET_NETWORK socket1851 6558 driver34c
SOCKET_NETWORK socket1852 6559 driver34d
SOCKET_NETWORK socket1853 6560 driver34e
SOCKET_NETWORK socket1854 6561 driver34f
SOCKET_NETWORK socket1855 6562 driver34a
SOCKET_NETWORK socket1856 6563 driver34b
SOCKET_NETWORK socket1857 6564 driver34c
SOCKET_NETWORK socket1858 6565 driver34d
SOCKET_NETWORK socket1859 6566 driver34e
SOCKET_NETWORK socket1860 6567 driver34f
SOCKET_NETWORK socket1861 6568 driver34a
SOCKET_NETWORK socket1862 6569 driver34b
SOCKET_NETWORK socket1863 6570 driver34c
SOCKET_NETWORK socket1864 6571 driver34d
SOCKET_NETWORK socket1865 6572 driver34e
SOCKET_NETWORK socket1866 6573 driver34f
SOCKET_NETWORK socket1867 6574 driver34a
SOCKET_NETWORK socket1868 6575 driver34b
SOCKET_NETWORK socket1869 6576 driver34c
SOCKET_NETWORK socket1870 6577 driver34d
SOCKET_NETWORK socket1871 6578 driver34e
SOCKET_NETWORK socket1872 6579 driver34f
SOCKET_NETWORK socket1873 6580 driver34a
SOCKET_NETWORK socket1874 6581 driver34b
SOCKET_NETWORK socket1875 6582 driver34c
SOCKET_NETWORK socket1876 6583 driver34d
SOCKET_NETWORK socket1877 6584 driver34e
SOCKET_NETWORK socket1878 6585 driver34f
SOCKET_NETWORK socket1879 6586 driver34a
SOCKET_NETWORK socket1880 6587 driver34b
SOCKET_NETWORK socket1881 6588 driver34c
SOCKET_NETWORK socket1882 6589 driver34d
SOCKET_NETWORK socket1883 6590 driver34e
SOCKET_NETWORK socket1884 6591 driver34f
SOCKET_NETWORK socket1885 6592 driver34a
SOCKET_NETWORK socket1886 6593 driver34b
SOCKET_NETWORK socket1887 6594 driver34c
SOCKET_NETWORK socket1888 6595 driver34d
SOCKET_NETWORK socket1889 6596 driver34e
SOCKET_NETWORK socket1890 6597 driver34f
SOCKET_NETWORK socket1891 6598 driver34a
SOCKET_NETWORK socket1892 6599 driver34b
SOCKET_NETWORK socket1893 6600 driver34c
SOCKET_NETWORK socket1894 6601 driver34d
SOCKET_NETWORK socket1895 6602 driver34e
SOCKET_NETWORK socket1896 6603 driver34f
SOCKET_NETWORK socket1897 6604 driver34a
SOCKET_NETWORK socket1898 6605 driver34b
SOCKET_NETWORK socket1899 6606 driver34c
SOCKET_NETWORK socket1900 6607 driver34d
SOCKET_NETWORK socket1901 6608 driver34e
SOCKET_NETWORK socket1902 6609 driver34f
SOCKET_NETWORK socket1903 6610 driver34a
SOCKET_NETWORK socket1904 6611 driver34b
SOCKET_NETWORK socket1905 6500 driver35a
SOCKET_NETWORK socket1906 6501 driver35b
SOCKET_NETWORK socket1907 6502 driver35c
SOCKET_NETWORK socket1908 6503 driver35d
SOCKET_NETWORK socket1909 6504 driver35e
SOCKET_NETWORK socket1910 6505 driver35f
SOCKET_NETWORK socket1911 6506 driver35a
SOCKET_NETWORK socket1912 6507 driver35b
SOCKET_NETWORK socket1913 6508 driver35c
SOCKET_NETWORK socket1914 6509 driver35d
SOCKET_NETWORK socket1915 6510 driver35e
SOCKET_NETWORK socket1916 6511 driver35f
SOCKET_NETWORK socket1917 6512 driver35a
SOCKET_NETWORK socket1918 6513 driver35b
SOCKET_NETWORK socket1919 6514 driver35c
SOCKET_NETWORK socket1920 6515 driver35d
SOCKET_NETWORK socket1921 6516 driver35e
SOCKET_NETWORK socket1922 6517 driver35f
SOCKET_NETWORK socket1923 6518 driver35a
SOCKET_NETWORK socket1924 6519 driver35b
SOCKET_NETWORK socket1925 6520 driver35c
SOCKET_NETWORK socket1926 6521 driver35d
SOCKET_NETWORK socket1927 6522 driver35e
SOCKET_NETWORK socket1928 6523 driver35f
SOCKET_NETWORK socket1929 6524 driver35a
SOCKET_NETWORK socket1930 6525 driver35b
SOCKET_NETWORK socket1931 6526 driver35c
SOCKET_NETWORK socket1932 6527 driver35d
SOCKET_NETWORK socket1933 6528 driver35e
SOCKET_NETWORK socket1934 6529 driver35f
SOCKET_NETWORK socket1935 6530 driver35a
SOCKET_NETWORK socket1936 6531 driver35b
SOCKET_NETWORK socket1937 6532 driver35c
SOCKET_NETWORK socket1938 6533 driver35d
SOCKET_NETWORK socket1939 6534 driver35e
SOCKET_NETWORK socket1940 6535 driver35f
SOCKET_NETWORK socket1941 6536 driver35a
SOCKET_NETWORK socket1942 6537 driver35b
SOCKET_NETWORK socket1943 6538 driver35c
SOCKET_NETWORK socket1944 6539 driver35d
SOCKET_NETWORK socket1945 6540 driver35e
SOCKET_NETWORK socket1946 6541 driver35f
SOCKET_NETWORK socket1947 6542 driver35a
SOCKET_NETWORK socket1948 6543 driver35b
SOCKET_NETWORK socket1949 6544 driver35c
SOCKET_NETWORK socket1950 6545 driver35d
SOCKET_NETWORK socket1951 6546 driver35e
SOCKET_NETWORK socket1952 6547 driver35f
SOCKET_NETWORK socket1953 6548 driver35a
SOCKET_NETWORK socket1954 6549 driver35b
SOCKET_NETWORK socket1955 6550 driver35c
SOCKET_NETWORK socket1956 6551 driver35d
SOCKET_NETWORK socket1957 6552 driver35e
SOCKET_NETWORK socket1958 6553 driver35f
SOCKET_NETWORK socket1959 6554 driver35a
SOCKET_NETWORK socket1960 6555 driver35b
SOCKET_NETWORK socket1961 6556 driver36a
SOCKET_NETWORK socket1962 6557 driver36b
SOCKET_NETWORK socket1963 6558 driver36c
SOCKET_NETWORK socket1964 6559 driver36d
SOCKET_NETWORK socket1965 6560 driver36e
SOCKET_NETWORK socket1966 6561 driver36f
SOCKET_NETWORK socket1967 6562 driver36a
SOCKET_NETWORK socket1968 6563 driver36b
SOCKET_NETWORK socket1969 6564 driver36c
SOCKET_NETWORK socket1970 6565 driver36d
SOCKET_NETWORK socket1971 6566 driver36e
SOCKET_NETWORK socket1972 6567 driver36f
SOCKET_NETWORK socket1973 6568 driver36a
SOCKET_NETWORK socket1974 6569 driver36b
SOCKET_NETWORK socket1975 6570 driver36c
SOCKET_NETWORK socket1976 6571 driver36d
SOCKET_NETWORK socket1977 6572 driver36e
SOCKET_NETWORK socket1978 6573 driver36f
SOCKET_NETWORK socket1979 6574 driver36a
SOCKET_NETWORK socket1980 6575 driver36b
SOCKET_NETWORK socket1981 6576 driver36c
SOCKET_NETWORK socket1982 6577 driver36d
SOCKET_NETWORK socket1983 6578 driver36e
SOCKET_NETWORK socket1984 6579 driver36f
SOCKET_NETWORK socket1985 6580 driver36a
SOCKET_NETWORK socket1986 6581 driver36b
SOCKET_NETWORK socket1987 6582 driver36c
SOCKET_NETWORK socket1988 6583 driver36d
SOCKET_NETWORK socket1989 6584 driver36e
SOCKET_NETWORK socket1990 6585 driver36f
SOCKET_NETWORK socket1991 6586 driver36a
SOCKET_NETWORK socket1992 6587 driver36b
SOCKET_NETWORK socket1993 6588 driver36c
SOCKET_NETWORK socket1994 6589 driver36d
SOCKET_NETWORK socket1995 6590 driver36e
SOCKET_NETWORK socket1996 6591 driver36f
SOCKET_NETWORK socket1997 6592 driver36a
SOCKET_NETWORK socket1998 6593 driver36b
SOCKET_NETWORK socket1999 6594 driver36c
SOCKET_NETWORK socket2000 6595 driver36d
SOCKET_NETWORK socket2001 6596 driver36e
SOCKET_NETWORK socket2002 6597 driver36f
SOCKET_NETWORK socket2003 6598 driver36a
SOCKET_NETWORK socket2004 6599 driver36b
SOCKET_NETWORK socket2005 6600 driver36c
SOCKET_NETWORK socket2006 6601 driver36d
SOCKET_NETWORK socket2007 6602 driver36e
SOCKET_NETWORK socket2008 6603 driver36f
SOCKET_NETWORK socket2009 6604 driver36a
SOCKET_NETWORK socket2010 6605 driver36b
SOCKET_NETWORK socket2011 6606 driver36c
SOCKET_NETWORK socket2012 6607 driver36d
SOCKET_NETWORK socket2013 6608 driver36e
SOCKET_NETWORK socket2014 6609 driver36f
SOCKET_NETWORK socket2015 6610 driver36a
SOCKET_NETWORK socket2016 6611 driver36b
#elif MASTER_NUM7
SOCKET_NETWORK socket2017 6500 driver37a
SOCKET_NETWORK socket2018 6501 driver37b
SOCKET_NETWORK socket2019 6502 driver37c
SOCKET_NETWORK socket2020 6503 driver37d
SOCKET_NETWORK socket2021 6504 driver37e
SOCKET_NETWORK socket2022 6505 driver37f
SOCKET_NETWORK socket2023 6506 driver37a
SOCKET_NETWORK socket2024 6507 driver37b
SOCKET_NETWORK socket2025 6508 driver37c
SOCKET_NETWORK socket2026 6509 driver37d
SOCKET_NETWORK socket2027 6510 driver37e
SOCKET_NETWORK socket2028 6511 driver37f
SOCKET_NETWORK socket2029 6512 driver37a
SOCKET_NETWORK socket2030 6513 driver37b
SOCKET_NETWORK socket2031 6514 driver37c
SOCKET_NETWORK socket2032 6515 driver37d
SOCKET_NETWORK socket2033 6516 driver37e
SOCKET_NETWORK socket2034 6517 driver37f

SOCKET_NETWORK socket2035 6518 driver37a
SOCKET_NETWORK socket2036 6519 driver37b
SOCKET_NETWORK socket2037 6520 driver37c
SOCKET_NETWORK socket2038 6521 driver37d
SOCKET_NETWORK socket2039 6522 driver37e
SOCKET_NETWORK socket2040 6523 driver37f
SOCKET_NETWORK socket2041 6524 driver37a
SOCKET_NETWORK socket2042 6525 driver37b
SOCKET_NETWORK socket2043 6526 driver37c
SOCKET_NETWORK socket2044 6527 driver37d
SOCKET_NETWORK socket2045 6528 driver37e
SOCKET_NETWORK socket2046 6529 driver37f
SOCKET_NETWORK socket2047 6530 driver37a
SOCKET_NETWORK socket2048 6531 driver37b
SOCKET_NETWORK socket2049 6532 driver37c
SOCKET_NETWORK socket2050 6533 driver37d
SOCKET_NETWORK socket2051 6534 driver37e
SOCKET_NETWORK socket2052 6535 driver37f
SOCKET_NETWORK socket2053 6536 driver37a
SOCKET_NETWORK socket2054 6537 driver37b
SOCKET_NETWORK socket2055 6538 driver37c
SOCKET_NETWORK socket2056 6539 driver37d
SOCKET_NETWORK socket2057 6540 driver37e
SOCKET_NETWORK socket2058 6541 driver37f
SOCKET_NETWORK socket2059 6542 driver37a
SOCKET_NETWORK socket2060 6543 driver37b
SOCKET_NETWORK socket2061 6544 driver37c
SOCKET_NETWORK socket2062 6545 driver37d
SOCKET_NETWORK socket2063 6546 driver37e
SOCKET_NETWORK socket2064 6547 driver37f
SOCKET_NETWORK socket2065 6548 driver37a
SOCKET_NETWORK socket2066 6549 driver37b
SOCKET_NETWORK socket2067 6550 driver37c
SOCKET_NETWORK socket2068 6551 driver37d
SOCKET_NETWORK socket2069 6552 driver37e
SOCKET_NETWORK socket2070 6553 driver37f
SOCKET_NETWORK socket2071 6554 driver37a
SOCKET_NETWORK socket2072 6555 driver37b
SOCKET_NETWORK socket2073 6556 driver38a
SOCKET_NETWORK socket2074 6557 driver38b
SOCKET_NETWORK socket2075 6558 driver38c
SOCKET_NETWORK socket2076 6559 driver38d
SOCKET_NETWORK socket2077 6560 driver38e
SOCKET_NETWORK socket2078 6561 driver38f
SOCKET_NETWORK socket2079 6562 driver38a
SOCKET_NETWORK socket2080 6563 driver38b
SOCKET_NETWORK socket2081 6564 driver38c
SOCKET_NETWORK socket2082 6565 driver38d
SOCKET_NETWORK socket2083 6566 driver38e
SOCKET_NETWORK socket2084 6567 driver38f
SOCKET_NETWORK socket2085 6568 driver38a
SOCKET_NETWORK socket2086 6569 driver38b
SOCKET_NETWORK socket2087 6570 driver38c
SOCKET_NETWORK socket2088 6571 driver38d
SOCKET_NETWORK socket2089 6572 driver38e
SOCKET_NETWORK socket2090 6573 driver38f
SOCKET_NETWORK socket2091 6574 driver38a
SOCKET_NETWORK socket2092 6575 driver38b
SOCKET_NETWORK socket2093 6576 driver38c
SOCKET_NETWORK socket2094 6577 driver38d
SOCKET_NETWORK socket2095 6578 driver38e
SOCKET_NETWORK socket2096 6579 driver38f
SOCKET_NETWORK socket2097 6580 driver38a
SOCKET_NETWORK socket2098 6581 driver38b
SOCKET_NETWORK socket2099 6582 driver38c
SOCKET_NETWORK socket2100 6583 driver38d
SOCKET_NETWORK socket2101 6584 driver38e
SOCKET_NETWORK socket2102 6585 driver38f
SOCKET_NETWORK socket2103 6586 driver38a
SOCKET_NETWORK socket2104 6587 driver38b
SOCKET_NETWORK socket2105 6588 driver38c
SOCKET_NETWORK socket2106 6589 driver38d
SOCKET_NETWORK socket2107 6590 driver38e
SOCKET_NETWORK socket2108 6591 driver38f
SOCKET_NETWORK socket2109 6592 driver38a
SOCKET_NETWORK socket2110 6593 driver38b
SOCKET_NETWORK socket2111 6594 driver38c
SOCKET_NETWORK socket2112 6595 driver38d
SOCKET_NETWORK socket2113 6596 driver38e
SOCKET_NETWORK socket2114 6597 driver38f
SOCKET_NETWORK socket2115 6598 driver38a
SOCKET_NETWORK socket2116 6599 driver38b
SOCKET_NETWORK socket2117 6600 driver38c
SOCKET_NETWORK socket2118 6601 driver38d
SOCKET_NETWORK socket2119 6602 driver38e
SOCKET_NETWORK socket2120 6603 driver38f
SOCKET_NETWORK socket2121 6604 driver38a
SOCKET_NETWORK socket2122 6605 driver38b
SOCKET_NETWORK socket2123 6606 driver38c
SOCKET_NETWORK socket2124 6607 driver38d
SOCKET_NETWORK socket2125 6608 driver38e
SOCKET_NETWORK socket2126 6609 driver38f
SOCKET_NETWORK socket2127 6610 driver38a
SOCKET_NETWORK socket2128 6611 driver38b
SOCKET_NETWORK socket2129 6500 driver39a
SOCKET_NETWORK socket2130 6501 driver39b
SOCKET_NETWORK socket2131 6502 driver39c
SOCKET_NETWORK socket2132 6503 driver39d
SOCKET_NETWORK socket2133 6504 driver39e
SOCKET_NETWORK socket2134 6505 driver39f
SOCKET_NETWORK socket2135 6506 driver39a
SOCKET_NETWORK socket2136 6507 driver39b
SOCKET_NETWORK socket2137 6508 driver39c
SOCKET_NETWORK socket2138 6509 driver39d
SOCKET_NETWORK socket2139 6510 driver39e
SOCKET_NETWORK socket2140 6511 driver39f
SOCKET_NETWORK socket2141 6512 driver39a
SOCKET_NETWORK socket2142 6513 driver39b
SOCKET_NETWORK socket2143 6514 driver39c
SOCKET_NETWORK socket2144 6515 driver39d
SOCKET_NETWORK socket2145 6516 driver39e
SOCKET_NETWORK socket2146 6517 driver39f
SOCKET_NETWORK socket2147 6518 driver39a
SOCKET_NETWORK socket2148 6519 driver39b
SOCKET_NETWORK socket2149 6520 driver39c
SOCKET_NETWORK socket2150 6521 driver39d
SOCKET_NETWORK socket2151 6522 driver39e
SOCKET_NETWORK socket2152 6523 driver39f
SOCKET_NETWORK socket2153 6524 driver39a
SOCKET_NETWORK socket2154 6525 driver39b
SOCKET_NETWORK socket2155 6526 driver39c
SOCKET_NETWORK socket2156 6527 driver39d
SOCKET_NETWORK socket2157 6528 driver39e
SOCKET_NETWORK socket2158 6529 driver39f
SOCKET_NETWORK socket2159 6530 driver39a
SOCKET_NETWORK socket2160 6531 driver39b
SOCKET_NETWORK socket2161 6532 driver39c
SOCKET_NETWORK socket2162 6533 driver39d
SOCKET_NETWORK socket2163 6534 driver39e
SOCKET_NETWORK socket2164 6535 driver39f
SOCKET_NETWORK socket2165 6536 driver39a
SOCKET_NETWORK socket2166 6537 driver39b
SOCKET_NETWORK socket2167 6538 driver39c
SOCKET_NETWORK socket2168 6539 driver39d
SOCKET_NETWORK socket2169 6540 driver39e
SOCKET_NETWORK socket2170 6541 driver39f
SOCKET_NETWORK socket2171 6542 driver39a
SOCKET_NETWORK socket2172 6543 driver39b
SOCKET_NETWORK socket2173 6544 driver39c
SOCKET_NETWORK socket2174 6545 driver39d
SOCKET_NETWORK socket2175 6546 driver39e
SOCKET_NETWORK socket2176 6547 driver39f
SOCKET_NETWORK socket2177 6548 driver39a
SOCKET_NETWORK socket2178 6549 driver39b
SOCKET_NETWORK socket2179 6550 driver39c
SOCKET_NETWORK socket2180 6551 driver39d
SOCKET_NETWORK socket2181 6552 driver39e
SOCKET_NETWORK socket2182 6553 driver39f
SOCKET_NETWORK socket2183 6554 driver39a
SOCKET_NETWORK socket2184 6555 driver39b
SOCKET_NETWORK socket2185 6556 driver40a
SOCKET_NETWORK socket2186 6557 driver40b
SOCKET_NETWORK socket2187 6558 driver40c
SOCKET_NETWORK socket2188 6559 driver40d
SOCKET_NETWORK socket2189 6560 driver40e
SOCKET_NETWORK socket2190 6561 driver40f
SOCKET_NETWORK socket2191 6562 driver40a
SOCKET_NETWORK socket2192 6563 driver40b
SOCKET_NETWORK socket2193 6564 driver40c
SOCKET_NETWORK socket2194 6565 driver40d
SOCKET_NETWORK socket2195 6566 driver40e
SOCKET_NETWORK socket2196 6567 driver40f
SOCKET_NETWORK socket2197 6568 driver40a
SOCKET_NETWORK socket2198 6569 driver40b
SOCKET_NETWORK socket2199 6570 driver40c
SOCKET_NETWORK socket2200 6571 driver40d
SOCKET_NETWORK socket2201 6572 driver40e
SOCKET_NETWORK socket2202 6573 driver40f
SOCKET_NETWORK socket2203 6574 driver40a
SOCKET_NETWORK socket2204 6575 driver40b
SOCKET_NETWORK socket2205 6576 driver40c
SOCKET_NETWORK socket2206 6577 driver40d
SOCKET_NETWORK socket2207 6578 driver40e
SOCKET_NETWORK socket2208 6579 driver40f
SOCKET_NETWORK socket2209 6580 driver40a
SOCKET_NETWORK socket2210 6581 driver40b
SOCKET_NETWORK socket2211 6582 driver40c
SOCKET_NETWORK socket2212 6583 driver40d
SOCKET_NETWORK socket2213 6584 driver40e
SOCKET_NETWORK socket2214 6585 driver40f
SOCKET_NETWORK socket2215 6586 driver40a
SOCKET_NETWORK socket2216 6587 driver40b
SOCKET_NETWORK socket2217 6588 driver40c
SOCKET_NETWORK socket2218 6589 driver40d
SOCKET_NETWORK socket2219 6590 driver40e
SOCKET_NETWORK socket2220 6591 driver40f
SOCKET_NETWORK socket2221 6592 driver40a
SOCKET_NETWORK socket2222 6593 driver40b
SOCKET_NETWORK socket2223 6594 driver40c
SOCKET_NETWORK socket2224 6595 driver40d
SOCKET_NETWORK socket2225 6596 driver40e
SOCKET_NETWORK socket2226 6597 driver40f
SOCKET_NETWORK socket2227 6598 driver40a
SOCKET_NETWORK socket2228 6599 driver40b
SOCKET_NETWORK socket2229 6600 driver40c
SOCKET_NETWORK socket2230 6601 driver40d
SOCKET_NETWORK socket2231 6602 driver40e
SOCKET_NETWORK socket2232 6603 driver40f

SOCKET_NETWORK socket2233 6604 driver40a
SOCKET_NETWORK socket2234 6605 driver40b
SOCKET_NETWORK socket2235 6606 driver40c
SOCKET_NETWORK socket2236 6607 driver40d
SOCKET_NETWORK socket2237 6608 driver40e
SOCKET_NETWORK socket2238 6609 driver40f
SOCKET_NETWORK socket2239 6610 driver40a
SOCKET_NETWORK socket2240 6611 driver40b
SOCKET_NETWORK socket2241 6500 driver41a
SOCKET_NETWORK socket2242 6501 driver41b
SOCKET_NETWORK socket2243 6502 driver41c
SOCKET_NETWORK socket2244 6503 driver41d
SOCKET_NETWORK socket2245 6504 driver41e
SOCKET_NETWORK socket2246 6505 driver41f
SOCKET_NETWORK socket2247 6506 driver41a
SOCKET_NETWORK socket2248 6507 driver41b
SOCKET_NETWORK socket2249 6508 driver41c
SOCKET_NETWORK socket2250 6509 driver41d
SOCKET_NETWORK socket2251 6510 driver41e
SOCKET_NETWORK socket2252 6511 driver41f
SOCKET_NETWORK socket2253 6512 driver41a
SOCKET_NETWORK socket2254 6513 driver41b
SOCKET_NETWORK socket2255 6514 driver41c
SOCKET_NETWORK socket2256 6515 driver41d
SOCKET_NETWORK socket2257 6516 driver41e
SOCKET_NETWORK socket2258 6517 driver41f
SOCKET_NETWORK socket2259 6518 driver41a
SOCKET_NETWORK socket2260 6519 driver41b
SOCKET_NETWORK socket2261 6520 driver41c
SOCKET_NETWORK socket2262 6521 driver41d
SOCKET_NETWORK socket2263 6522 driver41e
SOCKET_NETWORK socket2264 6523 driver41f
SOCKET_NETWORK socket2265 6524 driver41a
SOCKET_NETWORK socket2266 6525 driver41b
SOCKET_NETWORK socket2267 6526 driver41c
SOCKET_NETWORK socket2268 6527 driver41d
SOCKET_NETWORK socket2269 6528 driver41e
SOCKET_NETWORK socket2270 6529 driver41f
SOCKET_NETWORK socket2271 6530 driver41a
SOCKET_NETWORK socket2272 6531 driver41b
SOCKET_NETWORK socket2273 6532 driver41c
SOCKET_NETWORK socket2274 6533 driver41d
SOCKET_NETWORK socket2275 6534 driver41e
SOCKET_NETWORK socket2276 6535 driver41f
SOCKET_NETWORK socket2277 6536 driver41a
SOCKET_NETWORK socket2278 6537 driver41b
SOCKET_NETWORK socket2279 6538 driver41c
SOCKET_NETWORK socket2280 6539 driver41d
SOCKET_NETWORK socket2281 6540 driver41e
SOCKET_NETWORK socket2282 6541 driver41f
SOCKET_NETWORK socket2283 6542 driver41a
SOCKET_NETWORK socket2284 6543 driver41b
SOCKET_NETWORK socket2285 6544 driver41c
SOCKET_NETWORK socket2286 6545 driver41d
SOCKET_NETWORK socket2287 6546 driver41e
SOCKET_NETWORK socket2288 6547 driver41f
SOCKET_NETWORK socket2289 6548 driver41a
SOCKET_NETWORK socket2290 6549 driver41b
SOCKET_NETWORK socket2291 6550 driver41c
SOCKET_NETWORK socket2292 6551 driver41d
SOCKET_NETWORK socket2293 6552 driver41e
SOCKET_NETWORK socket2294 6553 driver41f
SOCKET_NETWORK socket2295 6554 driver41a
SOCKET_NETWORK socket2296 6555 driver41b
SOCKET_NETWORK socket2297 6556 driver42a
SOCKET_NETWORK socket2298 6557 driver42b
SOCKET_NETWORK socket2299 6558 driver42c
SOCKET_NETWORK socket2300 6559 driver42d
SOCKET_NETWORK socket2301 6560 driver42e
SOCKET_NETWORK socket2302 6561 driver42f
SOCKET_NETWORK socket2303 6562 driver42a
SOCKET_NETWORK socket2304 6563 driver42b
SOCKET_NETWORK socket2305 6564 driver42c
SOCKET_NETWORK socket2306 6565 driver42d
SOCKET_NETWORK socket2307 6566 driver42e
SOCKET_NETWORK socket2308 6567 driver42f
SOCKET_NETWORK socket2309 6568 driver42a
SOCKET_NETWORK socket2310 6569 driver42b
SOCKET_NETWORK socket2311 6570 driver42c
SOCKET_NETWORK socket2312 6571 driver42d
SOCKET_NETWORK socket2313 6572 driver42e
SOCKET_NETWORK socket2314 6573 driver42f
SOCKET_NETWORK socket2315 6574 driver42a
SOCKET_NETWORK socket2316 6575 driver42b
SOCKET_NETWORK socket2317 6576 driver42c
SOCKET_NETWORK socket2318 6577 driver42d
SOCKET_NETWORK socket2319 6578 driver42e
SOCKET_NETWORK socket2320 6579 driver42f
SOCKET_NETWORK socket2321 6580 driver42a
SOCKET_NETWORK socket2322 6581 driver42b
SOCKET_NETWORK socket2323 6582 driver42c
SOCKET_NETWORK socket2324 6583 driver42d
SOCKET_NETWORK socket2325 6584 driver42e
SOCKET_NETWORK socket2326 6585 driver42f
SOCKET_NETWORK socket2327 6586 driver42a
SOCKET_NETWORK socket2328 6587 driver42b
SOCKET_NETWORK socket2329 6588 driver42c
SOCKET_NETWORK socket2330 6589 driver42d
SOCKET_NETWORK socket2331 6590 driver42e
SOCKET_NETWORK socket2332 6591 driver42f
SOCKET_NETWORK socket2333 6592 driver42a
SOCKET_NETWORK socket2334 6593 driver42b
SOCKET_NETWORK socket2335 6594 driver42c
SOCKET_NETWORK socket2336 6595 driver42d
SOCKET_NETWORK socket2337 6596 driver42e
SOCKET_NETWORK socket2338 6597 driver42f
SOCKET_NETWORK socket2339 6598 driver42a
SOCKET_NETWORK socket2340 6599 driver42b
SOCKET_NETWORK socket2341 6600 driver42c
SOCKET_NETWORK socket2342 6601 driver42d
SOCKET_NETWORK socket2343 6602 driver42e
SOCKET_NETWORK socket2344 6603 driver42f
SOCKET_NETWORK socket2345 6604 driver42a
SOCKET_NETWORK socket2346 6605 driver42b
SOCKET_NETWORK socket2347 6606 driver42c
SOCKET_NETWORK socket2348 6607 driver42d
SOCKET_NETWORK socket2349 6608 driver42e
SOCKET_NETWORK socket2350 6609 driver42f
SOCKET_NETWORK socket2351 6610 driver42a
SOCKET_NETWORK socket2352 6611 driver42b
#elif MASTER_NUM8
SOCKET_NETWORK socket2353 6500 driver43a
SOCKET_NETWORK socket2354 6501 driver43b
SOCKET_NETWORK socket2355 6502 driver43c
SOCKET_NETWORK socket2356 6503 driver43d
SOCKET_NETWORK socket2357 6504 driver43e
SOCKET_NETWORK socket2358 6505 driver43f
SOCKET_NETWORK socket2359 6506 driver43a
SOCKET_NETWORK socket2360 6507 driver43b
SOCKET_NETWORK socket2361 6508 driver43c
SOCKET_NETWORK socket2362 6509 driver43d
SOCKET_NETWORK socket2363 6510 driver43e
SOCKET_NETWORK socket2364 6511 driver43f
SOCKET_NETWORK socket2365 6512 driver43a
SOCKET_NETWORK socket2366 6513 driver43b
SOCKET_NETWORK socket2367 6514 driver43c
SOCKET_NETWORK socket2368 6515 driver43d
SOCKET_NETWORK socket2369 6516 driver43e
SOCKET_NETWORK socket2370 6517 driver43f
SOCKET_NETWORK socket2371 6518 driver43a
SOCKET_NETWORK socket2372 6519 driver43b
SOCKET_NETWORK socket2373 6520 driver43c
SOCKET_NETWORK socket2374 6521 driver43d
SOCKET_NETWORK socket2375 6522 driver43e
SOCKET_NETWORK socket2376 6523 driver43f
SOCKET_NETWORK socket2377 6524 driver43a
SOCKET_NETWORK socket2378 6525 driver43b
SOCKET_NETWORK socket2379 6526 driver43c
SOCKET_NETWORK socket2380 6527 driver43d
SOCKET_NETWORK socket2381 6528 driver43e
SOCKET_NETWORK socket2382 6529 driver43f
SOCKET_NETWORK socket2383 6530 driver43a
SOCKET_NETWORK socket2384 6531 driver43b
SOCKET_NETWORK socket2385 6532 driver43c
SOCKET_NETWORK socket2386 6533 driver43d
SOCKET_NETWORK socket2387 6534 driver43e
SOCKET_NETWORK socket2388 6535 driver43f
SOCKET_NETWORK socket2389 6536 driver43a
SOCKET_NETWORK socket2390 6537 driver43b
SOCKET_NETWORK socket2391 6538 driver43c
SOCKET_NETWORK socket2392 6539 driver43d
SOCKET_NETWORK socket2393 6540 driver43e
SOCKET_NETWORK socket2394 6541 driver43f
SOCKET_NETWORK socket2395 6542 driver43a
SOCKET_NETWORK socket2396 6543 driver43b
SOCKET_NETWORK socket2397 6544 driver43c
SOCKET_NETWORK socket2398 6545 driver43d
SOCKET_NETWORK socket2399 6546 driver43e
SOCKET_NETWORK socket2400 6547 driver43f
SOCKET_NETWORK socket2401 6548 driver43a
SOCKET_NETWORK socket2402 6549 driver43b
SOCKET_NETWORK socket2403 6550 driver43c
SOCKET_NETWORK socket2404 6551 driver43d
SOCKET_NETWORK socket2405 6552 driver43e
SOCKET_NETWORK socket2406 6553 driver43f
SOCKET_NETWORK socket2407 6554 driver43a
SOCKET_NETWORK socket2408 6555 driver43b
SOCKET_NETWORK socket2409 6556 driver44a
SOCKET_NETWORK socket2410 6557 driver44b
SOCKET_NETWORK socket2411 6558 driver44c
SOCKET_NETWORK socket2412 6559 driver44d
SOCKET_NETWORK socket2413 6560 driver44e
SOCKET_NETWORK socket2414 6561 driver44f
SOCKET_NETWORK socket2415 6562 driver44a
SOCKET_NETWORK socket2416 6563 driver44b
SOCKET_NETWORK socket2417 6564 driver44c
SOCKET_NETWORK socket2418 6565 driver44d
SOCKET_NETWORK socket2419 6566 driver44e
SOCKET_NETWORK socket2420 6567 driver44f
SOCKET_NETWORK socket2421 6568 driver44a
SOCKET_NETWORK socket2422 6569 driver44b
SOCKET_NETWORK socket2423 6570 driver44c
SOCKET_NETWORK socket2424 6571 driver44d
SOCKET_NETWORK socket2425 6572 driver44e
SOCKET_NETWORK socket2426 6573 driver44f
SOCKET_NETWORK socket2427 6574 driver44a
SOCKET_NETWORK socket2428 6575 driver44b
SOCKET_NETWORK socket2429 6576 driver44c

SOCKET_NETWORK socket2430 6577 driver44d
SOCKET_NETWORK socket2431 6578 driver44e
SOCKET_NETWORK socket2432 6579 driver44f
SOCKET_NETWORK socket2433 6580 driver44a
SOCKET_NETWORK socket2434 6581 driver44b
SOCKET_NETWORK socket2435 6582 driver44c
SOCKET_NETWORK socket2436 6583 driver44d
SOCKET_NETWORK socket2437 6584 driver44e
SOCKET_NETWORK socket2438 6585 driver44f
SOCKET_NETWORK socket2439 6586 driver44a
SOCKET_NETWORK socket2440 6587 driver44b
SOCKET_NETWORK socket2441 6588 driver44c
SOCKET_NETWORK socket2442 6589 driver44d
SOCKET_NETWORK socket2443 6590 driver44e
SOCKET_NETWORK socket2444 6591 driver44f
SOCKET_NETWORK socket2445 6592 driver44a
SOCKET_NETWORK socket2446 6593 driver44b
SOCKET_NETWORK socket2447 6594 driver44c
SOCKET_NETWORK socket2448 6595 driver44d
SOCKET_NETWORK socket2449 6596 driver44e
SOCKET_NETWORK socket2450 6597 driver44f
SOCKET_NETWORK socket2451 6598 driver44a
SOCKET_NETWORK socket2452 6599 driver44b
SOCKET_NETWORK socket2453 6600 driver44c
SOCKET_NETWORK socket2454 6601 driver44d
SOCKET_NETWORK socket2455 6602 driver44e
SOCKET_NETWORK socket2456 6603 driver44f
SOCKET_NETWORK socket2457 6604 driver44a
SOCKET_NETWORK socket2458 6605 driver44b
SOCKET_NETWORK socket2459 6606 driver44c
SOCKET_NETWORK socket2460 6607 driver44d
SOCKET_NETWORK socket2461 6608 driver44e
SOCKET_NETWORK socket2462 6609 driver44f
SOCKET_NETWORK socket2463 6610 driver44a
SOCKET_NETWORK socket2464 6611 driver44b
SOCKET_NETWORK socket2465 6500 driver45a
SOCKET_NETWORK socket2466 6501 driver45b
SOCKET_NETWORK socket2467 6502 driver45c
SOCKET_NETWORK socket2468 6503 driver45d
SOCKET_NETWORK socket2469 6504 driver45e
SOCKET_NETWORK socket2470 6505 driver45f
SOCKET_NETWORK socket2471 6506 driver45a
SOCKET_NETWORK socket2472 6507 driver45b
SOCKET_NETWORK socket2473 6508 driver45c
SOCKET_NETWORK socket2474 6509 driver45d
SOCKET_NETWORK socket2475 6510 driver45e
SOCKET_NETWORK socket2476 6511 driver45f
SOCKET_NETWORK socket2477 6512 driver45a
SOCKET_NETWORK socket2478 6513 driver45b
SOCKET_NETWORK socket2479 6514 driver45c
SOCKET_NETWORK socket2480 6515 driver45d
SOCKET_NETWORK socket2481 6516 driver45e
SOCKET_NETWORK socket2482 6517 driver45f
SOCKET_NETWORK socket2483 6518 driver45a
SOCKET_NETWORK socket2484 6519 driver45b
SOCKET_NETWORK socket2485 6520 driver45c
SOCKET_NETWORK socket2486 6521 driver45d
SOCKET_NETWORK socket2487 6522 driver45e
SOCKET_NETWORK socket2488 6523 driver45f
SOCKET_NETWORK socket2489 6524 driver45a
SOCKET_NETWORK socket2490 6525 driver45b
SOCKET_NETWORK socket2491 6526 driver45c
SOCKET_NETWORK socket2492 6527 driver45d
SOCKET_NETWORK socket2493 6528 driver45e
SOCKET_NETWORK socket2494 6529 driver45f
SOCKET_NETWORK socket2495 6530 driver45a
SOCKET_NETWORK socket2496 6531 driver45b
SOCKET_NETWORK socket2497 6532 driver45c
SOCKET_NETWORK socket2498 6533 driver45d
SOCKET_NETWORK socket2499 6534 driver45e
SOCKET_NETWORK socket2500 6535 driver45f
SOCKET_NETWORK socket2501 6536 driver45a
SOCKET_NETWORK socket2502 6537 driver45b
SOCKET_NETWORK socket2503 6538 driver45c
SOCKET_NETWORK socket2504 6539 driver45d
SOCKET_NETWORK socket2505 6540 driver45e
SOCKET_NETWORK socket2506 6541 driver45f
SOCKET_NETWORK socket2507 6542 driver45a
SOCKET_NETWORK socket2508 6543 driver45b
SOCKET_NETWORK socket2509 6544 driver45c
SOCKET_NETWORK socket2510 6545 driver45d
SOCKET_NETWORK socket2511 6546 driver45e
SOCKET_NETWORK socket2512 6547 driver45f
SOCKET_NETWORK socket2513 6548 driver45a
SOCKET_NETWORK socket2514 6549 driver45b
SOCKET_NETWORK socket2515 6550 driver45c
SOCKET_NETWORK socket2516 6551 driver45d
SOCKET_NETWORK socket2517 6552 driver45e
SOCKET_NETWORK socket2518 6553 driver45f
SOCKET_NETWORK socket2519 6554 driver45a
SOCKET_NETWORK socket2520 6555 driver45b
SOCKET_NETWORK socket2521 6556 driver46a
SOCKET_NETWORK socket2522 6557 driver46b
SOCKET_NETWORK socket2523 6558 driver46c
SOCKET_NETWORK socket2524 6559 driver46d
SOCKET_NETWORK socket2525 6560 driver46e
SOCKET_NETWORK socket2526 6561 driver46f
SOCKET_NETWORK socket2527 6562 driver46a
SOCKET_NETWORK socket2528 6563 driver46b
SOCKET_NETWORK socket2529 6564 driver46c
SOCKET_NETWORK socket2530 6565 driver46d
SOCKET_NETWORK socket2531 6566 driver46e
SOCKET_NETWORK socket2532 6567 driver46f
SOCKET_NETWORK socket2533 6568 driver46a
SOCKET_NETWORK socket2534 6569 driver46b
SOCKET_NETWORK socket2535 6570 driver46c
SOCKET_NETWORK socket2536 6571 driver46d
SOCKET_NETWORK socket2537 6572 driver46e
SOCKET_NETWORK socket2538 6573 driver46f
SOCKET_NETWORK socket2539 6574 driver46a
SOCKET_NETWORK socket2540 6575 driver46b
SOCKET_NETWORK socket2541 6576 driver46c
SOCKET_NETWORK socket2542 6577 driver46d
SOCKET_NETWORK socket2543 6578 driver46e
SOCKET_NETWORK socket2544 6579 driver46f
SOCKET_NETWORK socket2545 6580 driver46a
SOCKET_NETWORK socket2546 6581 driver46b
SOCKET_NETWORK socket2547 6582 driver46c
SOCKET_NETWORK socket2548 6583 driver46d
SOCKET_NETWORK socket2549 6584 driver46e
SOCKET_NETWORK socket2550 6585 driver46f
SOCKET_NETWORK socket2551 6586 driver46a
SOCKET_NETWORK socket2552 6587 driver46b
SOCKET_NETWORK socket2553 6588 driver46c
SOCKET_NETWORK socket2554 6589 driver46d
SOCKET_NETWORK socket2555 6590 driver46e
SOCKET_NETWORK socket2556 6591 driver46f
SOCKET_NETWORK socket2557 6592 driver46a
SOCKET_NETWORK socket2558 6593 driver46b
SOCKET_NETWORK socket2559 6594 driver46c
SOCKET_NETWORK socket2560 6595 driver46d
SOCKET_NETWORK socket2561 6596 driver46e
SOCKET_NETWORK socket2562 6597 driver46f
SOCKET_NETWORK socket2563 6598 driver46a
SOCKET_NETWORK socket2564 6599 driver46b
SOCKET_NETWORK socket2565 6600 driver46c
SOCKET_NETWORK socket2566 6601 driver46d
SOCKET_NETWORK socket2567 6602 driver46e
SOCKET_NETWORK socket2568 6603 driver46f
SOCKET_NETWORK socket2569 6604 driver46a
SOCKET_NETWORK socket2570 6605 driver46b
SOCKET_NETWORK socket2571 6606 driver46c
SOCKET_NETWORK socket2572 6607 driver46d
SOCKET_NETWORK socket2573 6608 driver46e
SOCKET_NETWORK socket2574 6609 driver46f
SOCKET_NETWORK socket2575 6610 driver46a
SOCKET_NETWORK socket2576 6611 driver46b
SOCKET_NETWORK socket2577 6500 driver47a
SOCKET_NETWORK socket2578 6501 driver47b
SOCKET_NETWORK socket2579 6502 driver47c
SOCKET_NETWORK socket2580 6503 driver47d
SOCKET_NETWORK socket2581 6504 driver47e
SOCKET_NETWORK socket2582 6505 driver47f
SOCKET_NETWORK socket2583 6506 driver47a
SOCKET_NETWORK socket2584 6507 driver47b
SOCKET_NETWORK socket2585 6508 driver47c
SOCKET_NETWORK socket2586 6509 driver47d
SOCKET_NETWORK socket2587 6510 driver47e
SOCKET_NETWORK socket2588 6511 driver47f
SOCKET_NETWORK socket2589 6512 driver47a
SOCKET_NETWORK socket2590 6513 driver47b
SOCKET_NETWORK socket2591 6514 driver47c
SOCKET_NETWORK socket2592 6515 driver47d
SOCKET_NETWORK socket2593 6516 driver47e
SOCKET_NETWORK socket2594 6517 driver47f
SOCKET_NETWORK socket2595 6518 driver47a
SOCKET_NETWORK socket2596 6519 driver47b
SOCKET_NETWORK socket2597 6520 driver47c
SOCKET_NETWORK socket2598 6521 driver47d
SOCKET_NETWORK socket2599 6522 driver47e
SOCKET_NETWORK socket2600 6523 driver47f
SOCKET_NETWORK socket2601 6524 driver47a
SOCKET_NETWORK socket2602 6525 driver47b
SOCKET_NETWORK socket2603 6526 driver47c
SOCKET_NETWORK socket2604 6527 driver47d
SOCKET_NETWORK socket2605 6528 driver47e
SOCKET_NETWORK socket2606 6529 driver47f
SOCKET_NETWORK socket2607 6530 driver47a
SOCKET_NETWORK socket2608 6531 driver47b
SOCKET_NETWORK socket2609 6532 driver47c
SOCKET_NETWORK socket2610 6533 driver47d
SOCKET_NETWORK socket2611 6534 driver47e
SOCKET_NETWORK socket2612 6535 driver47f
SOCKET_NETWORK socket2613 6536 driver47a
SOCKET_NETWORK socket2614 6537 driver47b
SOCKET_NETWORK socket2615 6538 driver47c
SOCKET_NETWORK socket2616 6539 driver47d
SOCKET_NETWORK socket2617 6540 driver47e
SOCKET_NETWORK socket2618 6541 driver47f
SOCKET_NETWORK socket2619 6542 driver47a
SOCKET_NETWORK socket2620 6543 driver47b
SOCKET_NETWORK socket2621 6544 driver47c
SOCKET_NETWORK socket2622 6545 driver47d
SOCKET_NETWORK socket2623 6546 driver47e
SOCKET_NETWORK socket2624 6547 driver47f
SOCKET_NETWORK socket2625 6548 driver47a
SOCKET_NETWORK socket2626 6549 driver47b
SOCKET_NETWORK socket2627 6550 driver47c

SOCKET_NETWORK socket2628 6551 driver47d
SOCKET_NETWORK socket2629 6552 driver47e
SOCKET_NETWORK socket2630 6553 driver47f
SOCKET_NETWORK socket2631 6554 driver47a
SOCKET_NETWORK socket2632 6555 driver47b
SOCKET_NETWORK socket2633 6556 driver48a
SOCKET_NETWORK socket2634 6557 driver48b
SOCKET_NETWORK socket2635 6558 driver48c
SOCKET_NETWORK socket2636 6559 driver48d
SOCKET_NETWORK socket2637 6560 driver48e
SOCKET_NETWORK socket2638 6561 driver48f
SOCKET_NETWORK socket2639 6562 driver48a
SOCKET_NETWORK socket2640 6563 driver48b
SOCKET_NETWORK socket2641 6564 driver48c
SOCKET_NETWORK socket2642 6565 driver48d
SOCKET_NETWORK socket2643 6566 driver48e
SOCKET_NETWORK socket2644 6567 driver48f
SOCKET_NETWORK socket2645 6568 driver48a
SOCKET_NETWORK socket2646 6569 driver48b
SOCKET_NETWORK socket2647 6570 driver48c
SOCKET_NETWORK socket2648 6571 driver48d
SOCKET_NETWORK socket2649 6572 driver48e
SOCKET_NETWORK socket2650 6573 driver48f
SOCKET_NETWORK socket2651 6574 driver48a
SOCKET_NETWORK socket2652 6575 driver48b
SOCKET_NETWORK socket2653 6576 driver48c
SOCKET_NETWORK socket2654 6577 driver48d
SOCKET_NETWORK socket2655 6578 driver48e
SOCKET_NETWORK socket2656 6579 driver48f
SOCKET_NETWORK socket2657 6580 driver48a
SOCKET_NETWORK socket2658 6581 driver48b
SOCKET_NETWORK socket2659 6582 driver48c
SOCKET_NETWORK socket2660 6583 driver48d
SOCKET_NETWORK socket2661 6584 driver48e
SOCKET_NETWORK socket2662 6585 driver48f
SOCKET_NETWORK socket2663 6586 driver48a
SOCKET_NETWORK socket2664 6587 driver48b
SOCKET_NETWORK socket2665 6588 driver48c
SOCKET_NETWORK socket2666 6589 driver48d
SOCKET_NETWORK socket2667 6590 driver48e
SOCKET_NETWORK socket2668 6591 driver48f
SOCKET_NETWORK socket2669 6592 driver48a
SOCKET_NETWORK socket2670 6593 driver48b
SOCKET_NETWORK socket2671 6594 driver48c
SOCKET_NETWORK socket2672 6595 driver48d
SOCKET_NETWORK socket2673 6596 driver48e
SOCKET_NETWORK socket2674 6597 driver48f
SOCKET_NETWORK socket2675 6598 driver48a
SOCKET_NETWORK socket2676 6599 driver48b
SOCKET_NETWORK socket2677 6600 driver48c
SOCKET_NETWORK socket2678 6601 driver48d
SOCKET_NETWORK socket2679 6602 driver48e
SOCKET_NETWORK socket2680 6603 driver48f
SOCKET_NETWORK socket2681 6604 driver48a
SOCKET_NETWORK socket2682 6605 driver48b
SOCKET_NETWORK socket2683 6606 driver48c
SOCKET_NETWORK socket2684 6607 driver48d
SOCKET_NETWORK socket2685 6608 driver48e
SOCKET_NETWORK socket2686 6609 driver48f
SOCKET_NETWORK socket2687 6610 driver48a
SOCKET_NETWORK socket2688 6611 driver48b
#elif MASTER_NUM9
SOCKET_NETWORK socket2689 6500 driver49a
SOCKET_NETWORK socket2690 6501 driver49b
SOCKET_NETWORK socket2691 6502 driver49c
SOCKET_NETWORK socket2692 6503 driver49d
SOCKET_NETWORK socket2693 6504 driver49e
SOCKET_NETWORK socket2694 6505 driver49f
SOCKET_NETWORK socket2695 6506 driver49a
SOCKET_NETWORK socket2696 6507 driver49b
SOCKET_NETWORK socket2697 6508 driver49c
SOCKET_NETWORK socket2698 6509 driver49d
SOCKET_NETWORK socket2699 6510 driver49e
SOCKET_NETWORK socket2700 6511 driver49f
SOCKET_NETWORK socket2701 6512 driver49a
SOCKET_NETWORK socket2702 6513 driver49b
SOCKET_NETWORK socket2703 6514 driver49c
SOCKET_NETWORK socket2704 6515 driver49d
SOCKET_NETWORK socket2705 6516 driver49e
SOCKET_NETWORK socket2706 6517 driver49f
SOCKET_NETWORK socket2707 6518 driver49a
SOCKET_NETWORK socket2708 6519 driver49b
SOCKET_NETWORK socket2709 6520 driver49c
SOCKET_NETWORK socket2710 6521 driver49d
SOCKET_NETWORK socket2711 6522 driver49e
SOCKET_NETWORK socket2712 6523 driver49f
SOCKET_NETWORK socket2713 6524 driver49a
SOCKET_NETWORK socket2714 6525 driver49b
SOCKET_NETWORK socket2715 6526 driver49c
SOCKET_NETWORK socket2716 6527 driver49d
SOCKET_NETWORK socket2717 6528 driver49e
SOCKET_NETWORK socket2718 6529 driver49f
SOCKET_NETWORK socket2719 6530 driver49a
SOCKET_NETWORK socket2720 6531 driver49b
SOCKET_NETWORK socket2721 6532 driver49c
SOCKET_NETWORK socket2722 6533 driver49d
SOCKET_NETWORK socket2723 6534 driver49e
SOCKET_NETWORK socket2724 6535 driver49f
SOCKET_NETWORK socket2725 6536 driver49a
SOCKET_NETWORK socket2726 6537 driver49b
SOCKET_NETWORK socket2727 6538 driver49c
SOCKET_NETWORK socket2728 6539 driver49d
SOCKET_NETWORK socket2729 6540 driver49e
SOCKET_NETWORK socket2730 6541 driver49f
SOCKET_NETWORK socket2731 6542 driver49a
SOCKET_NETWORK socket2732 6543 driver49b
SOCKET_NETWORK socket2733 6544 driver49c
SOCKET_NETWORK socket2734 6545 driver49d
SOCKET_NETWORK socket2735 6546 driver49e
SOCKET_NETWORK socket2736 6547 driver49f
SOCKET_NETWORK socket2737 6548 driver49a
SOCKET_NETWORK socket2738 6549 driver49b
SOCKET_NETWORK socket2739 6550 driver49c
SOCKET_NETWORK socket2740 6551 driver49d
SOCKET_NETWORK socket2741 6552 driver49e
SOCKET_NETWORK socket2742 6553 driver49f
SOCKET_NETWORK socket2743 6554 driver49a
SOCKET_NETWORK socket2744 6555 driver49b
SOCKET_NETWORK socket2745 6556 driver50a
SOCKET_NETWORK socket2746 6557 driver50b
SOCKET_NETWORK socket2747 6558 driver50c
SOCKET_NETWORK socket2748 6559 driver50d
SOCKET_NETWORK socket2749 6560 driver50e
SOCKET_NETWORK socket2750 6561 driver50f
SOCKET_NETWORK socket2751 6562 driver50a
SOCKET_NETWORK socket2752 6563 driver50b
SOCKET_NETWORK socket2753 6564 driver50c
SOCKET_NETWORK socket2754 6565 driver50d
SOCKET_NETWORK socket2755 6566 driver50e
SOCKET_NETWORK socket2756 6567 driver50f
SOCKET_NETWORK socket2757 6568 driver50a
SOCKET_NETWORK socket2758 6569 driver50b
SOCKET_NETWORK socket2759 6570 driver50c
SOCKET_NETWORK socket2760 6571 driver50d
SOCKET_NETWORK socket2761 6572 driver50e
SOCKET_NETWORK socket2762 6573 driver50f
SOCKET_NETWORK socket2763 6574 driver50a
SOCKET_NETWORK socket2764 6575 driver50b
SOCKET_NETWORK socket2765 6576 driver50c
SOCKET_NETWORK socket2766 6577 driver50d
SOCKET_NETWORK socket2767 6578 driver50e
SOCKET_NETWORK socket2768 6579 driver50f
SOCKET_NETWORK socket2769 6580 driver50a
SOCKET_NETWORK socket2770 6581 driver50b
SOCKET_NETWORK socket2771 6582 driver50c
SOCKET_NETWORK socket2772 6583 driver50d
SOCKET_NETWORK socket2773 6584 driver50e
SOCKET_NETWORK socket2774 6585 driver50f
SOCKET_NETWORK socket2775 6586 driver50a
SOCKET_NETWORK socket2776 6587 driver50b
SOCKET_NETWORK socket2777 6588 driver50c
SOCKET_NETWORK socket2778 6589 driver50d
SOCKET_NETWORK socket2779 6590 driver50e
SOCKET_NETWORK socket2780 6591 driver50f
SOCKET_NETWORK socket2781 6592 driver50a
SOCKET_NETWORK socket2782 6593 driver50b
SOCKET_NETWORK socket2783 6594 driver50c
SOCKET_NETWORK socket2784 6595 driver50d
SOCKET_NETWORK socket2785 6596 driver50e
SOCKET_NETWORK socket2786 6597 driver50f
SOCKET_NETWORK socket2787 6598 driver50a
SOCKET_NETWORK socket2788 6599 driver50b
SOCKET_NETWORK socket2789 6600 driver50c
SOCKET_NETWORK socket2790 6601 driver50d
SOCKET_NETWORK socket2791 6602 driver50e
SOCKET_NETWORK socket2792 6603 driver50f
SOCKET_NETWORK socket2793 6604 driver50a
SOCKET_NETWORK socket2794 6605 driver50b
SOCKET_NETWORK socket2795 6606 driver50c
SOCKET_NETWORK socket2796 6607 driver50d
SOCKET_NETWORK socket2797 6608 driver50e
SOCKET_NETWORK socket2798 6609 driver50f
SOCKET_NETWORK socket2799 6610 driver50a
SOCKET_NETWORK socket2800 6611 driver50b
SOCKET_NETWORK socket2801 6500 driver51a
SOCKET_NETWORK socket2802 6501 driver51b
SOCKET_NETWORK socket2803 6502 driver51c
SOCKET_NETWORK socket2804 6503 driver51d
SOCKET_NETWORK socket2805 6504 driver51e
SOCKET_NETWORK socket2806 6505 driver51f
SOCKET_NETWORK socket2807 6506 driver51a
SOCKET_NETWORK socket2808 6507 driver51b
SOCKET_NETWORK socket2809 6508 driver51c
SOCKET_NETWORK socket2810 6509 driver51d
SOCKET_NETWORK socket2811 6510 driver51e
SOCKET_NETWORK socket2812 6511 driver51f
SOCKET_NETWORK socket2813 6512 driver51a
SOCKET_NETWORK socket2814 6513 driver51b
SOCKET_NETWORK socket2815 6514 driver51c
SOCKET_NETWORK socket2816 6515 driver51d
SOCKET_NETWORK socket2817 6516 driver51e
SOCKET_NETWORK socket2818 6517 driver51f
SOCKET_NETWORK socket2819 6518 driver51a
SOCKET_NETWORK socket2820 6519 driver51b
SOCKET_NETWORK socket2821 6520 driver51c
SOCKET_NETWORK socket2822 6521 driver51d
SOCKET_NETWORK socket2823 6522 driver51e
SOCKET_NETWORK socket2824 6523 driver51f

SOCKET_NETWORK socket2825 6524 driver51a
SOCKET_NETWORK socket2826 6525 driver51b
SOCKET_NETWORK socket2827 6526 driver51c
SOCKET_NETWORK socket2828 6527 driver51d
SOCKET_NETWORK socket2829 6528 driver51e
SOCKET_NETWORK socket2830 6529 driver51f
SOCKET_NETWORK socket2831 6530 driver51a
SOCKET_NETWORK socket2832 6531 driver51b
SOCKET_NETWORK socket2833 6532 driver51c
SOCKET_NETWORK socket2834 6533 driver51d
SOCKET_NETWORK socket2835 6534 driver51e
SOCKET_NETWORK socket2836 6535 driver51f
SOCKET_NETWORK socket2837 6536 driver51a
SOCKET_NETWORK socket2838 6537 driver51b
SOCKET_NETWORK socket2839 6538 driver51c
SOCKET_NETWORK socket2840 6539 driver51d
SOCKET_NETWORK socket2841 6540 driver51e
SOCKET_NETWORK socket2842 6541 driver51f
SOCKET_NETWORK socket2843 6542 driver51a
SOCKET_NETWORK socket2844 6543 driver51b
SOCKET_NETWORK socket2845 6544 driver51c
SOCKET_NETWORK socket2846 6545 driver51d
SOCKET_NETWORK socket2847 6546 driver51e
SOCKET_NETWORK socket2848 6547 driver51f
SOCKET_NETWORK socket2849 6548 driver51a
SOCKET_NETWORK socket2850 6549 driver51b
SOCKET_NETWORK socket2851 6550 driver51c
SOCKET_NETWORK socket2852 6551 driver51d
SOCKET_NETWORK socket2853 6552 driver51e
SOCKET_NETWORK socket2854 6553 driver51f
SOCKET_NETWORK socket2855 6554 driver51a
SOCKET_NETWORK socket2856 6555 driver51b
SOCKET_NETWORK socket2857 6556 driver52a
SOCKET_NETWORK socket2858 6557 driver52b
SOCKET_NETWORK socket2859 6558 driver52c
SOCKET_NETWORK socket2860 6559 driver52d
SOCKET_NETWORK socket2861 6560 driver52e
SOCKET_NETWORK socket2862 6561 driver52f
SOCKET_NETWORK socket2863 6562 driver52a
SOCKET_NETWORK socket2864 6563 driver52b
SOCKET_NETWORK socket2865 6564 driver52c
SOCKET_NETWORK socket2866 6565 driver52d
SOCKET_NETWORK socket2867 6566 driver52e
SOCKET_NETWORK socket2868 6567 driver52f
SOCKET_NETWORK socket2869 6568 driver52a
SOCKET_NETWORK socket2870 6569 driver52b
SOCKET_NETWORK socket2871 6570 driver52c
SOCKET_NETWORK socket2872 6571 driver52d
SOCKET_NETWORK socket2873 6572 driver52e
SOCKET_NETWORK socket2874 6573 driver52f
SOCKET_NETWORK socket2875 6574 driver52a
SOCKET_NETWORK socket2876 6575 driver52b
SOCKET_NETWORK socket2877 6576 driver52c
SOCKET_NETWORK socket2878 6577 driver52d
SOCKET_NETWORK socket2879 6578 driver52e
SOCKET_NETWORK socket2880 6579 driver52f
SOCKET_NETWORK socket2881 6580 driver52a
SOCKET_NETWORK socket2882 6581 driver52b
SOCKET_NETWORK socket2883 6582 driver52c
SOCKET_NETWORK socket2884 6583 driver52d
SOCKET_NETWORK socket2885 6584 driver52e
SOCKET_NETWORK socket2886 6585 driver52f
SOCKET_NETWORK socket2887 6586 driver52a
SOCKET_NETWORK socket2888 6587 driver52b
SOCKET_NETWORK socket2889 6588 driver52c
SOCKET_NETWORK socket2890 6589 driver52d
SOCKET_NETWORK socket2891 6590 driver52e
SOCKET_NETWORK socket2892 6591 driver52f
SOCKET_NETWORK socket2893 6592 driver52a
SOCKET_NETWORK socket2894 6593 driver52b
SOCKET_NETWORK socket2895 6594 driver52c
SOCKET_NETWORK socket2896 6595 driver52d
SOCKET_NETWORK socket2897 6596 driver52e
SOCKET_NETWORK socket2898 6597 driver52f
SOCKET_NETWORK socket2899 6598 driver52a
SOCKET_NETWORK socket2900 6599 driver52b
SOCKET_NETWORK socket2901 6600 driver52c
SOCKET_NETWORK socket2902 6601 driver52d
SOCKET_NETWORK socket2903 6602 driver52e
SOCKET_NETWORK socket2904 6603 driver52f
SOCKET_NETWORK socket2905 6604 driver52a
SOCKET_NETWORK socket2906 6605 driver52b
SOCKET_NETWORK socket2907 6606 driver52c
SOCKET_NETWORK socket2908 6607 driver52d
SOCKET_NETWORK socket2909 6608 driver52e
SOCKET_NETWORK socket2910 6609 driver52f
SOCKET_NETWORK socket2911 6610 driver52a
SOCKET_NETWORK socket2912 6611 driver52b
SOCKET_NETWORK socket2913 6500 driver53a
SOCKET_NETWORK socket2914 6501 driver53b
SOCKET_NETWORK socket2915 6502 driver53c
SOCKET_NETWORK socket2916 6503 driver53d
SOCKET_NETWORK socket2917 6504 driver53e
SOCKET_NETWORK socket2918 6505 driver53f
SOCKET_NETWORK socket2919 6506 driver53a
SOCKET_NETWORK socket2920 6507 driver53b
SOCKET_NETWORK socket2921 6508 driver53c
SOCKET_NETWORK socket2922 6509 driver53d
SOCKET_NETWORK socket2923 6510 driver53e
SOCKET_NETWORK socket2924 6511 driver53f
SOCKET_NETWORK socket2925 6512 driver53a
SOCKET_NETWORK socket2926 6513 driver53b
SOCKET_NETWORK socket2927 6514 driver53c
SOCKET_NETWORK socket2928 6515 driver53d
SOCKET_NETWORK socket2929 6516 driver53e
SOCKET_NETWORK socket2930 6517 driver53f
SOCKET_NETWORK socket2931 6518 driver53a
SOCKET_NETWORK socket2932 6519 driver53b
SOCKET_NETWORK socket2933 6520 driver53c
SOCKET_NETWORK socket2934 6521 driver53d
SOCKET_NETWORK socket2935 6522 driver53e
SOCKET_NETWORK socket2936 6523 driver53f
SOCKET_NETWORK socket2937 6524 driver53a
SOCKET_NETWORK socket2938 6525 driver53b
SOCKET_NETWORK socket2939 6526 driver53c
SOCKET_NETWORK socket2940 6527 driver53d
SOCKET_NETWORK socket2941 6528 driver53e
SOCKET_NETWORK socket2942 6529 driver53f
SOCKET_NETWORK socket2943 6530 driver53a
SOCKET_NETWORK socket2944 6531 driver53b
SOCKET_NETWORK socket2945 6532 driver53c
SOCKET_NETWORK socket2946 6533 driver53d
SOCKET_NETWORK socket2947 6534 driver53e
SOCKET_NETWORK socket2948 6535 driver53f
SOCKET_NETWORK socket2949 6536 driver53a
SOCKET_NETWORK socket2950 6537 driver53b
SOCKET_NETWORK socket2951 6538 driver53c
SOCKET_NETWORK socket2952 6539 driver53d
SOCKET_NETWORK socket2953 6540 driver53e
SOCKET_NETWORK socket2954 6541 driver53f
SOCKET_NETWORK socket2955 6542 driver53a
SOCKET_NETWORK socket2956 6543 driver53b
SOCKET_NETWORK socket2957 6544 driver53c
SOCKET_NETWORK socket2958 6545 driver53d
SOCKET_NETWORK socket2959 6546 driver53e
SOCKET_NETWORK socket2960 6547 driver53f
SOCKET_NETWORK socket2961 6548 driver53a
SOCKET_NETWORK socket2962 6549 driver53b
SOCKET_NETWORK socket2963 6550 driver53c
SOCKET_NETWORK socket2964 6551 driver53d
SOCKET_NETWORK socket2965 6552 driver53e
SOCKET_NETWORK socket2966 6553 driver53f
SOCKET_NETWORK socket2967 6554 driver53a
SOCKET_NETWORK socket2968 6555 driver53b
SOCKET_NETWORK socket2969 6556 driver54a
SOCKET_NETWORK socket2970 6557 driver54b
SOCKET_NETWORK socket2971 6558 driver54c
SOCKET_NETWORK socket2972 6559 driver54d
SOCKET_NETWORK socket2973 6560 driver54e
SOCKET_NETWORK socket2974 6561 driver54f
SOCKET_NETWORK socket2975 6562 driver54a
SOCKET_NETWORK socket2976 6563 driver54b
SOCKET_NETWORK socket2977 6564 driver54c
SOCKET_NETWORK socket2978 6565 driver54d
SOCKET_NETWORK socket2979 6566 driver54e
SOCKET_NETWORK socket2980 6567 driver54f
SOCKET_NETWORK socket2981 6568 driver54a
SOCKET_NETWORK socket2982 6569 driver54b
SOCKET_NETWORK socket2983 6570 driver54c
SOCKET_NETWORK socket2984 6571 driver54d
SOCKET_NETWORK socket2985 6572 driver54e
SOCKET_NETWORK socket2986 6573 driver54f
SOCKET_NETWORK socket2987 6574 driver54a
SOCKET_NETWORK socket2988 6575 driver54b
SOCKET_NETWORK socket2989 6576 driver54c
SOCKET_NETWORK socket2990 6577 driver54d
SOCKET_NETWORK socket2991 6578 driver54e
SOCKET_NETWORK socket2992 6579 driver54f
SOCKET_NETWORK socket2993 6580 driver54a
SOCKET_NETWORK socket2994 6581 driver54b
SOCKET_NETWORK socket2995 6582 driver54c
SOCKET_NETWORK socket2996 6583 driver54d
SOCKET_NETWORK socket2997 6584 driver54e
SOCKET_NETWORK socket2998 6585 driver54f
SOCKET_NETWORK socket2999 6586 driver54a
SOCKET_NETWORK socket3000 6587 driver54b
SOCKET_NETWORK socket3001 6588 driver54c
SOCKET_NETWORK socket3002 6589 driver54d
SOCKET_NETWORK socket3003 6590 driver54e
SOCKET_NETWORK socket3004 6591 driver54f
SOCKET_NETWORK socket3005 6592 driver54a
SOCKET_NETWORK socket3006 6593 driver54b
SOCKET_NETWORK socket3007 6594 driver54c
SOCKET_NETWORK socket3008 6595 driver54d
SOCKET_NETWORK socket3009 6596 driver54e
SOCKET_NETWORK socket3010 6597 driver54f
SOCKET_NETWORK socket3011 6598 driver54a
SOCKET_NETWORK socket3012 6599 driver54b
SOCKET_NETWORK socket3013 6600 driver54c
SOCKET_NETWORK socket3014 6601 driver54d
SOCKET_NETWORK socket3015 6602 driver54e
SOCKET_NETWORK socket3016 6603 driver54f
SOCKET_NETWORK socket3017 6604 driver54a
SOCKET_NETWORK socket3018 6605 driver54b
SOCKET_NETWORK socket3019 6606 driver54c
SOCKET_NETWORK socket3020 6607 driver54d
SOCKET_NETWORK socket3021 6608 driver54e
SOCKET_NETWORK socket3022 6609 driver54f

```
SOCKET_NETWORK socket3023   6610 driver54a
SOCKET_NETWORK socket3024   6611 driver54b
#endif
/*-------------------------------------------------------------------*/
OUTPUTNAME="regattaH"
CPU=48
BEGIN_WAIT=15:00
RAMPUP=20:00
RUNTIME=30:00
RAMPDOWN_WAIT=5:00
RAMPDOWN=17:00
/*NOTE ** MUST SET CONVERSIONS AS WELL */
/*RAMPUP_SEC                1200 */
/*RUNTIME_SEC               1800 */
INTERVAL=1:00 /* Interval to calculate mix from */
LOGIN_MAX_LOAD = 4
LOGIN_BEGIN = 0 /* skip login state if set to 1 */
NOBEGIN = 1
KEYSTROKE_PACKET_SIZE = 0
MAX_CONCURRENT_SPAWN = 10
SPAWN_COUNT = 4
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 client1a socket1 170 0-17
START_RANGE client1b socket2 180 17-35
START_RANGE client1c socket3 170 35-52
START_RANGE client1d socket4 180 52-70
START_RANGE client1e socket5 170 70-87
START_RANGE client1f socket6 180 87-105
START_RANGE client1g socket7 180 105-123
START_RANGE client1h socket8 170 123-140
START_RANGE client1i socket9 180 140-158
START_RANGE client1j socket10 170 158-175
START_RANGE client1k socket11 180 175-193
START_RANGE client1l socket12 180 193-211
START_RANGE client1m socket13 170 211-228
START_RANGE client1n socket14 180 228-246
START_RANGE client1a socket15 170 246-263
START_RANGE client1b socket16 180 263-281
START_RANGE client1c socket17 180 281-299
START_RANGE client1d socket18 170 299-316
START_RANGE client1e socket19 180 316-334
START_RANGE client1f socket20 170 334-351
START_RANGE client1g socket21 180 351-369
START_RANGE client1h socket22 180 369-387
START_RANGE client1i socket23 170 387-404
START_RANGE client1j socket24 180 404-422
START_RANGE client1k socket25 170 422-439
START_RANGE client1l socket26 180 439-457
START_RANGE client1m socket27 170 457-474
START_RANGE client1n socket28 180 474-492
START_RANGE client1a socket29 180 492-510
START_RANGE client1b socket30 170 510-527
START_RANGE client1c socket31 180 527-545
START_RANGE client1d socket32 170 545-562
START_RANGE client1e socket33 180 562-580
START_RANGE client1f socket34 180 580-598
START_RANGE client1g socket35 170 598-615
START_RANGE client1h socket36 180 615-633
START_RANGE client1i socket37 170 633-650
START_RANGE client1j socket38 180 650-668
START_RANGE client1k socket39 180 668-686
START_RANGE client1l socket40 170 686-703
START_RANGE client1m socket41 180 703-721
START_RANGE client1n socket42 170 721-738
START_RANGE client1a socket43 180 738-756
START_RANGE client1b socket44 180 756-774
START_RANGE client1c socket45 170 774-791
START_RANGE client1d socket46 180 791-809
START_RANGE client1e socket47 170 809-826
START_RANGE client1f socket48 180 826-844
START_RANGE client1g socket49 180 844-862
START_RANGE client1h socket50 170 862-879
START_RANGE client1i socket51 180 879-897
START_RANGE client1j socket52 170 897-914
START_RANGE client1k socket53 180 914-932
START_RANGE client1l socket54 170 932-949
START_RANGE client1m socket55 180 949-967
START_RANGE client1n socket56 180 967-985

START_RANGE client1o socket57 170 985-1002
START_RANGE client1p socket58 180 1002-1020
START_RANGE client1q socket59 170 1020-1037
START_RANGE client1r socket60 180 1037-1055
START_RANGE client1s socket61 180 1055-1073
START_RANGE client1t socket62 170 1073-1090
START_RANGE client1u socket63 180 1090-1108
START_RANGE client1v socket64 170 1108-1125
START_RANGE client1w socket65 180 1125-1143
START_RANGE client1x socket66 180 1143-1161
START_RANGE client1y socket67 170 1161-1178
START_RANGE client1z socket68 180 1178-1196
START_RANGE client1aa socket69 170 1196-1213
START_RANGE client1ab socket70 180 1213-1231

START_RANGE client1o socket71 180 1231-1249
START_RANGE client1p socket72 170 1249-1266
START_RANGE client1q socket73 180 1266-1284
START_RANGE client1r socket74 170 1284-1301
START_RANGE client1s socket75 180 1301-1319
START_RANGE client1t socket76 180 1319-1337
START_RANGE client1u socket77 170 1337-1354
START_RANGE client1v socket78 180 1354-1372
START_RANGE client1w socket79 170 1372-1389
START_RANGE client1x socket80 180 1389-1407
START_RANGE client1y socket81 170 1407-1424
START_RANGE client1z socket82 180 1424-1442
START_RANGE client1aa socket83 180 1442-1460
START_RANGE client1ab socket84 170 1460-1477
START_RANGE client1o socket85 180 1477-1495
START_RANGE client1p socket86 170 1495-1512
START_RANGE client1q socket87 180 1512-1530
START_RANGE client1r socket88 180 1530-1548
START_RANGE client1s socket89 170 1548-1565
START_RANGE client1t socket90 180 1565-1583
START_RANGE client1u socket91 170 1583-1600
START_RANGE client1v socket92 180 1600-1618
START_RANGE client1w socket93 180 1618-1636
START_RANGE client1x socket94 170 1636-1653
START_RANGE client1y socket95 180 1653-1671
START_RANGE client1z socket96 170 1671-1688
START_RANGE client1aa socket97 180 1688-1706
START_RANGE client1ab socket98 180 1706-1724
START_RANGE client1o socket99 170 1724-1741
START_RANGE client1p socket100 180 1741-1759
START_RANGE client1q socket101 170 1759-1776
START_RANGE client1r socket102 180 1776-1794
START_RANGE client1s socket103 180 1794-1812
START_RANGE client1t socket104 170 1812-1829
START_RANGE client1u socket105 180 1829-1847
START_RANGE client1v socket106 170 1847-1864
START_RANGE client1w socket107 180 1864-1882
START_RANGE client1x socket108 170 1882-1899
START_RANGE client1y socket109 180 1899-1917
START_RANGE client1z socket110 180 1917-1935
START_RANGE client1aa socket111 170 1935-1952
START_RANGE client1ab socket112 180 1952-1970

START_RANGE client2a socket113 170 1970-1987
START_RANGE client2b socket114 180 1987-2005
START_RANGE client2c socket115 180 2005-2023
START_RANGE client2d socket116 170 2023-2040
START_RANGE client2e socket117 180 2040-2058
START_RANGE client2f socket118 170 2058-2075
START_RANGE client2g socket119 180 2075-2093
START_RANGE client2h socket120 180 2093-2111
START_RANGE client2i socket121 170 2111-2128
START_RANGE client2j socket122 180 2128-2146
START_RANGE client2k socket123 170 2146-2163
START_RANGE client2l socket124 180 2163-2181
START_RANGE client2m socket125 180 2181-2199
START_RANGE client2n socket126 170 2199-2216

START_RANGE client2a socket127 180 2216-2234
START_RANGE client2b socket128 170 2234-2251
START_RANGE client2c socket129 180 2251-2269
START_RANGE client2d socket130 180 2269-2287
START_RANGE client2e socket131 170 2287-2304
START_RANGE client2f socket132 180 2304-2322
START_RANGE client2g socket133 170 2322-2339
START_RANGE client2h socket134 180 2339-2357
START_RANGE client2i socket135 170 2357-2374
START_RANGE client2j socket136 180 2374-2392
START_RANGE client2k socket137 180 2392-2410
START_RANGE client2l socket138 170 2410-2427
START_RANGE client2m socket139 180 2427-2445
START_RANGE client2n socket140 170 2445-2462
START_RANGE client2a socket141 180 2462-2480
START_RANGE client2b socket142 180 2480-2498
START_RANGE client2c socket143 170 2498-2515
START_RANGE client2d socket144 180 2515-2533
START_RANGE client2e socket145 170 2533-2550
START_RANGE client2f socket146 180 2550-2568
START_RANGE client2g socket147 180 2568-2586
START_RANGE client2h socket148 170 2586-2603
START_RANGE client2i socket149 180 2603-2621
```

```
START_RANGE client2j socket150 170 2621-2638
START_RANGE client2k socket151 180 2638-2656
START_RANGE client2l socket152 180 2656-2674
START_RANGE client2m socket153 170 2674-2691
START_RANGE client2n socket154 180 2691-2709
START_RANGE client2a socket155 170 2709-2726
START_RANGE client2b socket156 180 2726-2744
START_RANGE client2c socket157 180 2744-2762
START_RANGE client2d socket158 170 2762-2779
START_RANGE client2e socket159 180 2779-2797
START_RANGE client2f socket160 170 2797-2814
START_RANGE client2g socket161 180 2814-2832
START_RANGE client2h socket162 170 2832-2849
START_RANGE client2i socket163 180 2849-2867
START_RANGE client2j socket164 180 2867-2885
START_RANGE client2k socket165 170 2885-2902
START_RANGE client2l socket166 180 2902-2920
START_RANGE client2m socket167 170 2920-2937
START_RANGE client2n socket168 180 2937-2955

START_RANGE client2o socket169 180 2955-2973
START_RANGE client2p socket170 170 2973-2990
START_RANGE client2q socket171 180 2990-3008
START_RANGE client2r socket172 170 3008-3025
START_RANGE client2s socket173 180 3025-3043
START_RANGE client2t socket174 180 3043-3061
START_RANGE client2u socket175 170 3061-3078
START_RANGE client2v socket176 180 3078-3096
START_RANGE client2w socket177 170 3096-3113
START_RANGE client2x socket178 180 3113-3131
START_RANGE client2y socket179 180 3131-3149
START_RANGE client2z socket180 170 3149-3166
START_RANGE client2aa socket181 180 3166-3184
START_RANGE client2ab socket182 170 3184-3201

START_RANGE client2o socket183 180 3201-3219
START_RANGE client2p socket184 180 3219-3237
START_RANGE client2q socket185 170 3237-3254
START_RANGE client2r socket186 180 3254-3272
START_RANGE client2s socket187 170 3272-3289
START_RANGE client2t socket188 180 3289-3307
START_RANGE client2u socket189 170 3307-3324
START_RANGE client2v socket190 180 3324-3342
START_RANGE client2w socket191 180 3342-3360
START_RANGE client2x socket192 170 3360-3377
START_RANGE client2y socket193 180 3377-3395
START_RANGE client2z socket194 170 3395-3412
START_RANGE client2aa socket195 180 3412-3430
START_RANGE client2ab socket196 180 3430-3448
START_RANGE client2o socket197 170 3448-3465
START_RANGE client2p socket198 180 3465-3483
START_RANGE client2q socket199 170 3483-3500
START_RANGE client2r socket200 180 3500-3518
START_RANGE client2s socket201 180 3518-3536
START_RANGE client2t socket202 170 3536-3553
START_RANGE client2u socket203 180 3553-3571
START_RANGE client2v socket204 170 3571-3588
START_RANGE client2w socket205 180 3588-3606
START_RANGE client2x socket206 180 3606-3624
START_RANGE client2y socket207 170 3624-3641
START_RANGE client2z socket208 180 3641-3659
START_RANGE client2aa socket209 170 3659-3676
START_RANGE client2ab socket210 180 3676-3694
START_RANGE client2o socket211 180 3694-3712
START_RANGE client2p socket212 170 3712-3729
START_RANGE client2q socket213 180 3729-3747
START_RANGE client2r socket214 170 3747-3764
START_RANGE client2s socket215 180 3764-3782
START_RANGE client2t socket216 170 3782-3799
START_RANGE client2u socket217 180 3799-3817
START_RANGE client2v socket218 180 3817-3835
START_RANGE client2w socket219 170 3835-3852
START_RANGE client2x socket220 180 3852-3870
START_RANGE client2y socket221 170 3870-3887
START_RANGE client2z socket222 180 3887-3905
START_RANGE client2aa socket223 180 3905-3923
START_RANGE client2ab socket224 170 3923-3940

START_RANGE client3a socket225 180 3940-3958
START_RANGE client3b socket226 170 3958-3975
START_RANGE client3c socket227 180 3975-3993
START_RANGE client3d socket228 180 3993-4011
START_RANGE client3e socket229 170 4011-4028
START_RANGE client3f socket230 180 4028-4046
START_RANGE client3g socket231 170 4046-4063
START_RANGE client3h socket232 180 4063-4081
START_RANGE client3i socket233 180 4081-4099
START_RANGE client3j socket234 170 4099-4116
START_RANGE client3k socket235 180 4116-4134
START_RANGE client3l socket236 170 4134-4151
START_RANGE client3m socket237 180 4151-4169
START_RANGE client3n socket238 180 4169-4187

START_RANGE client3a socket239 170 4187-4204
START_RANGE client3b socket240 180 4204-4222
START_RANGE client3c socket241 170 4222-4239
START_RANGE client3d socket242 180 4239-4257
START_RANGE client3e socket243 170 4257-4274
START_RANGE client3f socket244 180 4274-4292
START_RANGE client3g socket245 180 4292-4310
START_RANGE client3h socket246 170 4310-4327
START_RANGE client3i socket247 180 4327-4345
START_RANGE client3j socket248 170 4345-4362
START_RANGE client3k socket249 180 4362-4380
START_RANGE client3l socket250 180 4380-4398
START_RANGE client3m socket251 170 4398-4415
START_RANGE client3n socket252 180 4415-4433
START_RANGE client3a socket253 170 4433-4450
START_RANGE client3b socket254 180 4450-4468
START_RANGE client3c socket255 180 4468-4486
START_RANGE client3d socket256 170 4486-4503
START_RANGE client3e socket257 180 4503-4521
START_RANGE client3f socket258 170 4521-4538
START_RANGE client3g socket259 180 4538-4556
START_RANGE client3h socket260 180 4556-4574
START_RANGE client3i socket261 170 4574-4591
START_RANGE client3j socket262 180 4591-4609
START_RANGE client3k socket263 170 4609-4626
START_RANGE client3l socket264 180 4626-4644
START_RANGE client3m socket265 180 4644-4662
START_RANGE client3n socket266 170 4662-4679
START_RANGE client3a socket267 180 4679-4697
START_RANGE client3b socket268 170 4697-4714
START_RANGE client3c socket269 180 4714-4732
START_RANGE client3d socket270 170 4732-4749
START_RANGE client3e socket271 180 4749-4767
START_RANGE client3f socket272 180 4767-4785
START_RANGE client3g socket273 170 4785-4802
START_RANGE client3h socket274 180 4802-4820
START_RANGE client3i socket275 170 4820-4837
START_RANGE client3j socket276 180 4837-4855
START_RANGE client3k socket277 180 4855-4873
START_RANGE client3l socket278 170 4873-4890
START_RANGE client3m socket279 180 4890-4908
START_RANGE client3n socket280 170 4908-4925

START_RANGE client3o socket281 180 4925-4943
START_RANGE client3p socket282 180 4943-4961
START_RANGE client3q socket283 170 4961-4978
START_RANGE client3r socket284 180 4978-4996
START_RANGE client3s socket285 170 4996-5013
START_RANGE client3t socket286 180 5013-5031
START_RANGE client3u socket287 180 5031-5049
START_RANGE client3v socket288 170 5049-5066
START_RANGE client3w socket289 180 5066-5084
START_RANGE client3x socket290 170 5084-5101
START_RANGE client3y socket291 180 5101-5119
START_RANGE client3z socket292 180 5119-5137
START_RANGE client3aa socket293 170 5137-5154
START_RANGE client3ab socket294 180 5154-5172

START_RANGE client3o socket295 170 5172-5189
START_RANGE client3p socket296 180 5189-5207
START_RANGE client3q socket297 170 5207-5224
START_RANGE client3r socket298 180 5224-5242
START_RANGE client3s socket299 180 5242-5260
START_RANGE client3t socket300 170 5260-5277
START_RANGE client3u socket301 180 5277-5295
START_RANGE client3v socket302 170 5295-5312
START_RANGE client3w socket303 180 5312-5330
START_RANGE client3x socket304 180 5330-5348
START_RANGE client3y socket305 170 5348-5365
START_RANGE client3z socket306 180 5365-5383
START_RANGE client3aa socket307 170 5383-5400
START_RANGE client3ab socket308 180 5400-5418
START_RANGE client3o socket309 180 5418-5436
START_RANGE client3p socket310 170 5436-5453
START_RANGE client3q socket311 180 5453-5471
START_RANGE client3r socket312 170 5471-5488
START_RANGE client3s socket313 180 5488-5506
START_RANGE client3t socket314 180 5506-5524
START_RANGE client3u socket315 170 5524-5541
START_RANGE client3v socket316 180 5541-5559
START_RANGE client3w socket317 170 5559-5576
START_RANGE client3x socket318 180 5576-5594
START_RANGE client3y socket319 180 5594-5612
START_RANGE client3z socket320 170 5612-5629
START_RANGE client3aa socket321 180 5629-5647
START_RANGE client3ab socket322 170 5647-5664
START_RANGE client3o socket323 180 5664-5682
START_RANGE client3p socket324 170 5682-5699
START_RANGE client3q socket325 180 5699-5717
START_RANGE client3r socket326 180 5717-5735
START_RANGE client3s socket327 170 5735-5752
START_RANGE client3t socket328 180 5752-5770
START_RANGE client3u socket329 170 5770-5787
START_RANGE client3v socket330 180 5787-5805
START_RANGE client3w socket331 180 5805-5823
START_RANGE client3x socket332 170 5823-5840
START_RANGE client3y socket333 180 5840-5858
START_RANGE client3z socket334 170 5858-5875
START_RANGE client3aa socket335 180 5875-5893
START_RANGE client3ab socket336 180 5893-5911

#elif MASTER_NUM2
START_RANGE client4a socket337 170 5911-5928
START_RANGE client4b socket338 180 5928-5946
START_RANGE client4c socket339 170 5946-5963
```

START_RANGE client4d socket340 180 5963-5981
START_RANGE client4e socket341 180 5981-5999
START_RANGE client4f socket342 170 5999-6016
START_RANGE client4g socket343 180 6016-6034
START_RANGE client4h socket344 170 6034-6051
START_RANGE client4i socket345 180 6051-6069
START_RANGE client4j socket346 180 6069-6087
START_RANGE client4k socket347 170 6087-6104
START_RANGE client4l socket348 180 6104-6122
START_RANGE client4m socket349 170 6122-6139
START_RANGE client4n socket350 180 6139-6157
START_RANGE client4a socket351 170 6157-6174
START_RANGE client4b socket352 180 6174-6192
START_RANGE client4c socket353 180 6192-6210
START_RANGE client4d socket354 170 6210-6227
START_RANGE client4e socket355 180 6227-6245
START_RANGE client4f socket356 170 6245-6262
START_RANGE client4g socket357 180 6262-6280
START_RANGE client4h socket358 180 6280-6298
START_RANGE client4i socket359 170 6298-6315
START_RANGE client4j socket360 180 6315-6333
START_RANGE client4k socket361 170 6333-6350
START_RANGE client4l socket362 180 6350-6368
START_RANGE client4m socket363 180 6368-6386
START_RANGE client4n socket364 170 6386-6403
START_RANGE client4a socket365 180 6403-6421
START_RANGE client4b socket366 170 6421-6438
START_RANGE client4c socket367 180 6438-6456
START_RANGE client4d socket368 180 6456-6474
START_RANGE client4e socket369 170 6474-6491
START_RANGE client4f socket370 180 6491-6509
START_RANGE client4g socket371 170 6509-6526
START_RANGE client4h socket372 180 6526-6544
START_RANGE client4i socket373 180 6544-6562
START_RANGE client4j socket374 170 6562-6579
START_RANGE client4k socket375 180 6579-6597
START_RANGE client4l socket376 170 6597-6614
START_RANGE client4m socket377 180 6614-6632
START_RANGE client4n socket378 170 6632-6649
START_RANGE client4a socket379 180 6649-6667
START_RANGE client4b socket380 180 6667-6685
START_RANGE client4c socket381 170 6685-6702
START_RANGE client4d socket382 180 6702-6720
START_RANGE client4e socket383 170 6720-6737
START_RANGE client4f socket384 180 6737-6755
START_RANGE client4g socket385 180 6755-6773
START_RANGE client4h socket386 170 6773-6790
START_RANGE client4i socket387 180 6790-6808
START_RANGE client4j socket388 170 6808-6825
START_RANGE client4k socket389 180 6825-6843
START_RANGE client4l socket390 180 6843-6861
START_RANGE client4m socket391 170 6861-6878
START_RANGE client4n socket392 180 6878-6896

START_RANGE client4o socket393 170 6896-6913
START_RANGE client4p socket394 180 6913-6931
START_RANGE client4q socket395 180 6931-6949
START_RANGE client4r socket396 170 6949-6966
START_RANGE client4s socket397 180 6966-6984
START_RANGE client4t socket398 170 6984-7001
START_RANGE client4u socket399 180 7001-7019
START_RANGE client4v socket400 180 7019-7037
START_RANGE client4w socket401 170 7037-7054
START_RANGE client4x socket402 180 7054-7072
START_RANGE client4y socket403 170 7072-7089
START_RANGE client4z socket404 180 7089-7107
START_RANGE client4aa socket405 170 7107-7124
START_RANGE client4ab socket406 180 7124-7142
START_RANGE client4o socket407 180 7142-7160
START_RANGE client4p socket408 170 7160-7177
START_RANGE client4q socket409 180 7177-7195
START_RANGE client4r socket410 170 7195-7212
START_RANGE client4s socket411 180 7212-7230
START_RANGE client4t socket412 180 7230-7248
START_RANGE client4u socket413 170 7248-7265
START_RANGE client4v socket414 180 7265-7283
START_RANGE client4w socket415 170 7283-7300
START_RANGE client4x socket416 180 7300-7318
START_RANGE client4y socket417 180 7318-7336
START_RANGE client4z socket418 170 7336-7353
START_RANGE client4aa socket419 180 7353-7371
START_RANGE client4ab socket420 170 7371-7388
START_RANGE client4o socket421 180 7388-7406
START_RANGE client4p socket422 180 7406-7424
START_RANGE client4q socket423 170 7424-7441
START_RANGE client4r socket424 180 7441-7459
START_RANGE client4s socket425 170 7459-7476
START_RANGE client4t socket426 180 7476-7494
START_RANGE client4u socket427 180 7494-7512
START_RANGE client4v socket428 170 7512-7529
START_RANGE client4w socket429 180 7529-7547
START_RANGE client4x socket430 170 7547-7564
START_RANGE client4y socket431 180 7564-7582
START_RANGE client4z socket432 170 7582-7599
START_RANGE client4aa socket433 180 7599-7617
START_RANGE client4ab socket434 180 7617-7635
START_RANGE client4o socket435 170 7635-7652
START_RANGE client4p socket436 180 7652-7670
START_RANGE client4q socket437 170 7670-7687

START_RANGE client4r socket438 180 7687-7705
START_RANGE client4s socket439 180 7705-7723
START_RANGE client4t socket440 170 7723-7740
START_RANGE client4u socket441 180 7740-7758
START_RANGE client4v socket442 170 7758-7775
START_RANGE client4w socket443 180 7775-7793
START_RANGE client4x socket444 180 7793-7811
START_RANGE client4y socket445 170 7811-7828
START_RANGE client4z socket446 180 7828-7846
START_RANGE client4aa socket447 170 7846-7863
START_RANGE client4ab socket448 180 7863-7881

START_RANGE client5a socket449 180 7881-7899
START_RANGE client5b socket450 170 7899-7916
START_RANGE client5c socket451 180 7916-7934
START_RANGE client5d socket452 170 7934-7951
START_RANGE client5e socket453 180 7951-7969
START_RANGE client5f socket454 180 7969-7987
START_RANGE client5g socket455 170 7987-8004
START_RANGE client5h socket456 180 8004-8022
START_RANGE client5i socket457 170 8022-8039
START_RANGE client5j socket458 180 8039-8057
START_RANGE client5k socket459 170 8057-8074
START_RANGE client5l socket460 180 8074-8092
START_RANGE client5m socket461 180 8092-8110
START_RANGE client5n socket462 170 8110-8127
START_RANGE client5a socket463 180 8127-8145
START_RANGE client5b socket464 170 8145-8162
START_RANGE client5c socket465 180 8162-8180
START_RANGE client5d socket466 180 8180-8198
START_RANGE client5e socket467 170 8198-8215
START_RANGE client5f socket468 180 8215-8233
START_RANGE client5g socket469 170 8233-8250
START_RANGE client5h socket470 180 8250-8268
START_RANGE client5i socket471 180 8268-8286
START_RANGE client5j socket472 170 8286-8303
START_RANGE client5k socket473 180 8303-8321
START_RANGE client5l socket474 170 8321-8338
START_RANGE client5m socket475 180 8338-8356
START_RANGE client5n socket476 180 8356-8374
START_RANGE client5a socket477 170 8374-8391
START_RANGE client5b socket478 180 8391-8409
START_RANGE client5c socket479 170 8409-8426
START_RANGE client5d socket480 180 8426-8444
START_RANGE client5e socket481 180 8444-8462
START_RANGE client5f socket482 170 8462-8479
START_RANGE client5g socket483 180 8479-8497
START_RANGE client5h socket484 170 8497-8514
START_RANGE client5i socket485 180 8514-8532
START_RANGE client5j socket486 170 8532-8549
START_RANGE client5k socket487 180 8549-8567
START_RANGE client5l socket488 180 8567-8585
START_RANGE client5m socket489 170 8585-8602
START_RANGE client5n socket490 180 8602-8620
START_RANGE client5a socket491 170 8620-8637
START_RANGE client5b socket492 180 8637-8655
START_RANGE client5c socket493 180 8655-8673
START_RANGE client5d socket494 170 8673-8690
START_RANGE client5e socket495 180 8690-8708
START_RANGE client5f socket496 170 8708-8725
START_RANGE client5g socket497 180 8725-8743
START_RANGE client5h socket498 180 8743-8761
START_RANGE client5i socket499 170 8761-8778
START_RANGE client5j socket500 180 8778-8796
START_RANGE client5k socket501 170 8796-8813
START_RANGE client5l socket502 180 8813-8831
START_RANGE client5m socket503 180 8831-8849
START_RANGE client5n socket504 170 8849-8866

START_RANGE client5o socket505 180 8866-8884
START_RANGE client5p socket506 170 8884-8901
START_RANGE client5q socket507 180 8901-8919
START_RANGE client5r socket508 180 8919-8937
START_RANGE client5s socket509 170 8937-8954
START_RANGE client5t socket510 180 8954-8972
START_RANGE client5u socket511 170 8972-8989
START_RANGE client5v socket512 180 8989-9007
START_RANGE client5w socket513 170 9007-9024
START_RANGE client5x socket514 180 9024-9042
START_RANGE client5y socket515 180 9042-9060
START_RANGE client5z socket516 170 9060-9077
START_RANGE client5aa socket517 180 9077-9095
START_RANGE client5ab socket518 170 9095-9112
START_RANGE client5o socket519 180 9112-9130
START_RANGE client5p socket520 180 9130-9148
START_RANGE client5q socket521 170 9148-9165
START_RANGE client5r socket522 180 9165-9183
START_RANGE client5s socket523 170 9183-9200
START_RANGE client5t socket524 180 9200-9218
START_RANGE client5u socket525 180 9218-9236
START_RANGE client5v socket526 170 9236-9253
START_RANGE client5w socket527 180 9253-9271
START_RANGE client5x socket528 170 9271-9288
START_RANGE client5y socket529 180 9288-9306
START_RANGE client5z socket530 180 9306-9324
START_RANGE client5aa socket531 170 9324-9341
START_RANGE client5ab socket532 180 9341-9359
START_RANGE client5o socket533 170 9359-9376
START_RANGE client5p socket534 180 9376-9394

```
START_RANGE client5q socket535 180 9394-9412
START_RANGE client5r socket536 170 9412-9429
START_RANGE client5s socket537 180 9429-9447
START_RANGE client5t socket538 170 9447-9464
START_RANGE client5u socket539 180 9464-9482
START_RANGE client5v socket540 170 9482-9499
START_RANGE client5w socket541 180 9499-9517
START_RANGE client5x socket542 180 9517-9535
START_RANGE client5y socket543 170 9535-9552
START_RANGE client5z socket544 180 9552-9570
START_RANGE client5aa socket545 170 9570-9587
START_RANGE client5ab socket546 180 9587-9605
START_RANGE client5o socket547 180 9605-9623
START_RANGE client5p socket548 170 9623-9640
START_RANGE client5q socket549 180 9640-9658
START_RANGE client5r socket550 170 9658-9675
START_RANGE client5s socket551 180 9675-9693
START_RANGE client5t socket552 180 9693-9711
START_RANGE client5u socket553 170 9711-9728
START_RANGE client5v socket554 180 9728-9746
START_RANGE client5w socket555 170 9746-9763
START_RANGE client5x socket556 180 9763-9781
START_RANGE client5y socket557 180 9781-9799
START_RANGE client5z socket558 170 9799-9816
START_RANGE client5aa socket559 180 9816-9834
START_RANGE client5ab socket560 170 9834-9851

START_RANGE client6a socket561 180 9851-9869
START_RANGE client6b socket562 180 9869-9887
START_RANGE client6c socket563 170 9887-9904
START_RANGE client6d socket564 180 9904-9922
START_RANGE client6e socket565 170 9922-9939
START_RANGE client6f socket566 180 9939-9957
START_RANGE client6g socket567 170 9957-9974
START_RANGE client6h socket568 180 9974-9992
START_RANGE client6i socket569 180 9992-10010
START_RANGE client6j socket570 170 10010-10027
START_RANGE client6k socket571 180 10027-10045
START_RANGE client6l socket572 170 10045-10062
START_RANGE client6m socket573 180 10062-10080
START_RANGE client6n socket574 180 10080-10098
START_RANGE client6a socket575 170 10098-10115
START_RANGE client6b socket576 180 10115-10133
START_RANGE client6c socket577 170 10133-10150
START_RANGE client6d socket578 180 10150-10168
START_RANGE client6e socket579 180 10168-10186
START_RANGE client6f socket580 170 10186-10203
START_RANGE client6g socket581 180 10203-10221
START_RANGE client6h socket582 170 10221-10238
START_RANGE client6i socket583 180 10238-10256
START_RANGE client6j socket584 180 10256-10274
START_RANGE client6k socket585 170 10274-10291
START_RANGE client6l socket586 180 10291-10309
START_RANGE client6m socket587 170 10309-10326
START_RANGE client6n socket588 180 10326-10344
START_RANGE client6a socket589 180 10344-10362
START_RANGE client6b socket590 170 10362-10379
START_RANGE client6c socket591 180 10379-10397
START_RANGE client6d socket592 170 10397-10414
START_RANGE client6e socket593 180 10414-10432
START_RANGE client6f socket594 170 10432-10449
START_RANGE client6g socket595 180 10449-10467
START_RANGE client6h socket596 180 10467-10485
START_RANGE client6i socket597 170 10485-10502
START_RANGE client6j socket598 180 10502-10520
START_RANGE client6k socket599 170 10520-10537
START_RANGE client6l socket600 180 10537-10555
START_RANGE client6m socket601 180 10555-10573
START_RANGE client6n socket602 170 10573-10590
START_RANGE client6a socket603 180 10590-10608
START_RANGE client6b socket604 170 10608-10625
START_RANGE client6c socket605 180 10625-10643
START_RANGE client6d socket606 180 10643-10661
START_RANGE client6e socket607 170 10661-10678
START_RANGE client6f socket608 180 10678-10696
START_RANGE client6g socket609 170 10696-10713
START_RANGE client6h socket610 180 10713-10731
START_RANGE client6i socket611 180 10731-10749
START_RANGE client6j socket612 170 10749-10766
START_RANGE client6k socket613 180 10766-10784
START_RANGE client6l socket614 170 10784-10801
START_RANGE client6m socket615 180 10801-10819
START_RANGE client6n socket616 180 10819-10837

START_RANGE client6o socket617 170 10837-10854
START_RANGE client6p socket618 180 10854-10872
START_RANGE client6q socket619 170 10872-10889
START_RANGE client6r socket620 180 10889-10907
START_RANGE client6s socket621 170 10907-10924
START_RANGE client6t socket622 180 10924-10942
START_RANGE client6u socket623 180 10942-10960
START_RANGE client6v socket624 170 10960-10977
START_RANGE client6w socket625 180 10977-10995
START_RANGE client6x socket626 170 10995-11012
START_RANGE client6y socket627 180 11012-11030
START_RANGE client6z socket628 180 11030-11048
START_RANGE client6aa socket629 170 11048-11065
START_RANGE client6ab socket630 180 11065-11083
START_RANGE client6o socket631 170 11083-11100
START_RANGE client6p socket632 180 11100-11118
START_RANGE client6q socket633 180 11118-11136
START_RANGE client6r socket634 170 11136-11153
START_RANGE client6s socket635 180 11153-11171
START_RANGE client6t socket636 170 11171-11188
START_RANGE client6u socket637 180 11188-11206
START_RANGE client6v socket638 180 11206-11224
START_RANGE client6w socket639 170 11224-11241
START_RANGE client6x socket640 180 11241-11259
START_RANGE client6y socket641 170 11259-11276
START_RANGE client6z socket642 180 11276-11294
START_RANGE client6aa socket643 180 11294-11312
START_RANGE client6ab socket644 170 11312-11329
START_RANGE client6o socket645 180 11329-11347
START_RANGE client6p socket646 170 11347-11364
START_RANGE client6q socket647 180 11364-11382
START_RANGE client6r socket648 170 11382-11399
START_RANGE client6s socket649 180 11399-11417
START_RANGE client6t socket650 180 11417-11435
START_RANGE client6u socket651 170 11435-11452
START_RANGE client6v socket652 180 11452-11470
START_RANGE client6w socket653 170 11470-11487
START_RANGE client6x socket654 180 11487-11505
START_RANGE client6y socket655 180 11505-11523
START_RANGE client6z socket656 170 11523-11540
START_RANGE client6aa socket657 180 11540-11558
START_RANGE client6ab socket658 170 11558-11575
START_RANGE client6o socket659 180 11575-11593
START_RANGE client6p socket660 180 11593-11611
START_RANGE client6q socket661 170 11611-11628
START_RANGE client6r socket662 180 11628-11646
START_RANGE client6s socket663 170 11646-11663
START_RANGE client6t socket664 180 11663-11681
START_RANGE client6u socket665 180 11681-11699
START_RANGE client6v socket666 170 11699-11716
START_RANGE client6w socket667 180 11716-11734
START_RANGE client6x socket668 170 11734-11751
START_RANGE client6y socket669 180 11751-11769
START_RANGE client6z socket670 180 11769-11787
START_RANGE client6aa socket671 170 11787-11804
START_RANGE client6ab socket672 180 11804-11822

#elif MASTER_NUM3
START_RANGE client7a socket673 170 11822-11839
START_RANGE client7b socket674 180 11839-11857
START_RANGE client7c socket675 170 11857-11874
START_RANGE client7d socket676 180 11874-11892
START_RANGE client7e socket677 180 11892-11910
START_RANGE client7f socket678 170 11910-11927
START_RANGE client7g socket679 180 11927-11945
START_RANGE client7h socket680 170 11945-11962
START_RANGE client7i socket681 180 11962-11980
START_RANGE client7j socket682 180 11980-11998
START_RANGE client7k socket683 170 11998-12015
START_RANGE client7l socket684 180 12015-12033
START_RANGE client7m socket685 170 12033-12050
START_RANGE client7n socket686 180 12050-12068
START_RANGE client7a socket687 180 12068-12086
START_RANGE client7b socket688 170 12086-12103
START_RANGE client7c socket689 180 12103-12121
START_RANGE client7d socket690 170 12121-12138
START_RANGE client7e socket691 180 12138-12156
START_RANGE client7f socket692 180 12156-12174
START_RANGE client7g socket693 170 12174-12191
START_RANGE client7h socket694 180 12191-12209
START_RANGE client7i socket695 170 12209-12226
START_RANGE client7j socket696 180 12226-12244
START_RANGE client7k socket697 180 12244-12262
START_RANGE client7l socket698 170 12262-12279
START_RANGE client7m socket699 180 12279-12297
START_RANGE client7n socket700 170 12297-12314
START_RANGE client7a socket701 180 12314-12332
START_RANGE client7b socket702 170 12332-12349
START_RANGE client7c socket703 180 12349-12367
START_RANGE client7d socket704 180 12367-12385
START_RANGE client7e socket705 170 12385-12402
START_RANGE client7f socket706 180 12402-12420
START_RANGE client7g socket707 170 12420-12437
START_RANGE client7h socket708 180 12437-12455
START_RANGE client7i socket709 180 12455-12473
START_RANGE client7j socket710 170 12473-12490
START_RANGE client7k socket711 180 12490-12508
START_RANGE client7l socket712 170 12508-12525
START_RANGE client7m socket713 180 12525-12543
START_RANGE client7n socket714 180 12543-12561
START_RANGE client7a socket715 170 12561-12578
START_RANGE client7b socket716 180 12578-12596
START_RANGE client7c socket717 170 12596-12613
START_RANGE client7d socket718 180 12613-12631
START_RANGE client7e socket719 180 12631-12649
START_RANGE client7f socket720 170 12649-12666
START_RANGE client7g socket721 180 12666-12684
START_RANGE client7h socket722 170 12684-12701
START_RANGE client7i socket723 180 12701-12719
START_RANGE client7j socket724 180 12719-12737
START_RANGE client7k socket725 170 12737-12754
START_RANGE client7l socket726 180 12754-12772
START_RANGE client7m socket727 170 12772-12789
START_RANGE client7n socket728 180 12789-12807
```

START_RANGE client7o socket729 170 12807-12824
START_RANGE client7p socket730 180 12824-12842
START_RANGE client7q socket731 180 12842-12860
START_RANGE client7r socket732 170 12860-12877
START_RANGE client7s socket733 180 12877-12895
START_RANGE client7t socket734 170 12895-12912
START_RANGE client7u socket735 180 12912-12930
START_RANGE client7v socket736 180 12930-12948
START_RANGE client7w socket737 170 12948-12965
START_RANGE client7x socket738 180 12965-12983
START_RANGE client7y socket739 170 12983-13000
START_RANGE client7z socket740 180 13000-13018
START_RANGE client7aa socket741 180 13018-13036
START_RANGE client7ab socket742 170 13036-13053
START_RANGE client7o socket743 180 13053-13071
START_RANGE client7p socket744 170 13071-13088
START_RANGE client7q socket745 180 13088-13106
START_RANGE client7r socket746 180 13106-13124
START_RANGE client7s socket747 170 13124-13141
START_RANGE client7t socket748 180 13141-13159
START_RANGE client7u socket749 170 13159-13176
START_RANGE client7v socket750 180 13176-13194
START_RANGE client7w socket751 180 13194-13212
START_RANGE client7x socket752 170 13212-13229
START_RANGE client7y socket753 180 13229-13247
START_RANGE client7z socket754 170 13247-13264
START_RANGE client7aa socket755 180 13264-13282
START_RANGE client7ab socket756 170 13282-13299
START_RANGE client7o socket757 180 13299-13317
START_RANGE client7p socket758 180 13317-13335
START_RANGE client7q socket759 170 13335-13352
START_RANGE client7r socket760 180 13352-13370
START_RANGE client7s socket761 170 13370-13387
START_RANGE client7t socket762 180 13387-13405
START_RANGE client7u socket763 180 13405-13423
START_RANGE client7v socket764 170 13423-13440
START_RANGE client7w socket765 180 13440-13458
START_RANGE client7x socket766 170 13458-13475
START_RANGE client7y socket767 180 13475-13493
START_RANGE client7z socket768 180 13493-13511
START_RANGE client7aa socket769 170 13511-13528
START_RANGE client7ab socket770 180 13528-13546
START_RANGE client7o socket771 170 13546-13563
START_RANGE client7p socket772 180 13563-13581
START_RANGE client7q socket773 180 13581-13599
START_RANGE client7r socket774 170 13599-13616
START_RANGE client7s socket775 180 13616-13634
START_RANGE client7t socket776 170 13634-13651
START_RANGE client7u socket777 180 13651-13669
START_RANGE client7v socket778 180 13669-13687
START_RANGE client7w socket779 170 13687-13704
START_RANGE client7x socket780 180 13704-13722
START_RANGE client7y socket781 170 13722-13739
START_RANGE client7z socket782 180 13739-13757
START_RANGE client7aa socket783 170 13757-13774
START_RANGE client7ab socket784 180 13774-13792

START_RANGE client8a socket785 180 13792-13810
START_RANGE client8b socket786 170 13810-13827
START_RANGE client8c socket787 180 13827-13845
START_RANGE client8d socket788 170 13845-13862
START_RANGE client8e socket789 180 13862-13880
START_RANGE client8f socket790 180 13880-13898
START_RANGE client8g socket791 170 13898-13915
START_RANGE client8h socket792 180 13915-13933
START_RANGE client8i socket793 170 13933-13950
START_RANGE client8j socket794 180 13950-13968
START_RANGE client8k socket795 180 13968-13986
START_RANGE client8l socket796 170 13986-14003
START_RANGE client8m socket797 180 14003-14021
START_RANGE client8n socket798 170 14021-14038
START_RANGE client8a socket799 180 14038-14056
START_RANGE client8b socket800 180 14056-14074
START_RANGE client8c socket801 170 14074-14091
START_RANGE client8d socket802 180 14091-14109
START_RANGE client8e socket803 170 14109-14126
START_RANGE client8f socket804 180 14126-14144
START_RANGE client8g socket805 180 14144-14162
START_RANGE client8h socket806 170 14162-14179
START_RANGE client8i socket807 180 14179-14197
START_RANGE client8j socket808 170 14197-14214
START_RANGE client8k socket809 180 14214-14232
START_RANGE client8l socket810 170 14232-14249
START_RANGE client8m socket811 180 14249-14267
START_RANGE client8n socket812 180 14267-14285
START_RANGE client8a socket813 170 14285-14302
START_RANGE client8b socket814 180 14302-14320
START_RANGE client8c socket815 170 14320-14337
START_RANGE client8d socket816 180 14337-14355
START_RANGE client8e socket817 180 14355-14373
START_RANGE client8f socket818 170 14373-14390
START_RANGE client8g socket819 180 14390-14408
START_RANGE client8h socket820 170 14408-14425
START_RANGE client8i socket821 180 14425-14443
START_RANGE client8j socket822 180 14443-14461
START_RANGE client8k socket823 170 14461-14478
START_RANGE client8l socket824 180 14478-14496
START_RANGE client8m socket825 170 14496-14513
START_RANGE client8n socket826 180 14513-14531
START_RANGE client8a socket827 180 14531-14549
START_RANGE client8b socket828 170 14549-14566
START_RANGE client8c socket829 180 14566-14584
START_RANGE client8d socket830 170 14584-14601
START_RANGE client8e socket831 180 14601-14619
START_RANGE client8f socket832 180 14619-14637
START_RANGE client8g socket833 170 14637-14654
START_RANGE client8h socket834 180 14654-14672
START_RANGE client8i socket835 170 14672-14689
START_RANGE client8j socket836 180 14689-14707
START_RANGE client8k socket837 170 14707-14724
START_RANGE client8l socket838 180 14724-14742
START_RANGE client8m socket839 180 14742-14760
START_RANGE client8n socket840 170 14760-14777

START_RANGE client8o socket841 180 14777-14795
START_RANGE client8p socket842 170 14795-14812
START_RANGE client8q socket843 180 14812-14830
START_RANGE client8r socket844 180 14830-14848
START_RANGE client8s socket845 170 14848-14865
START_RANGE client8t socket846 180 14865-14883
START_RANGE client8u socket847 170 14883-14900
START_RANGE client8v socket848 180 14900-14918
START_RANGE client8w socket849 180 14918-14936
START_RANGE client8x socket850 170 14936-14953
START_RANGE client8y socket851 180 14953-14971
START_RANGE client8z socket852 170 14971-14988
START_RANGE client8aa socket853 180 14988-15006
START_RANGE client8ab socket854 180 15006-15024
START_RANGE client8o socket855 170 15024-15041
START_RANGE client8p socket856 180 15041-15059
START_RANGE client8q socket857 170 15059-15076
START_RANGE client8r socket858 180 15076-15094
START_RANGE client8s socket859 180 15094-15112
START_RANGE client8t socket860 170 15112-15129
START_RANGE client8u socket861 180 15129-15147
START_RANGE client8v socket862 170 15147-15164
START_RANGE client8w socket863 180 15164-15182
START_RANGE client8x socket864 170 15182-15199
START_RANGE client8y socket865 180 15199-15217
START_RANGE client8z socket866 180 15217-15235
START_RANGE client8aa socket867 170 15235-15252
START_RANGE client8ab socket868 180 15252-15270
START_RANGE client8o socket869 170 15270-15287
START_RANGE client8p socket870 180 15287-15305
START_RANGE client8q socket871 180 15305-15323
START_RANGE client8r socket872 170 15323-15340
START_RANGE client8s socket873 180 15340-15358
START_RANGE client8t socket874 170 15358-15375
START_RANGE client8u socket875 180 15375-15393
START_RANGE client8v socket876 180 15393-15411
START_RANGE client8w socket877 170 15411-15428
START_RANGE client8x socket878 180 15428-15446
START_RANGE client8y socket879 170 15446-15463
START_RANGE client8z socket880 180 15463-15481
START_RANGE client8aa socket881 180 15481-15499
START_RANGE client8ab socket882 170 15499-15516
START_RANGE client8o socket883 180 15516-15534
START_RANGE client8p socket884 170 15534-15551
START_RANGE client8q socket885 180 15551-15569
START_RANGE client8r socket886 180 15569-15587
START_RANGE client8s socket887 170 15587-15604
START_RANGE client8t socket888 180 15604-15622
START_RANGE client8u socket889 170 15622-15639
START_RANGE client8v socket890 180 15639-15657
START_RANGE client8w socket891 170 15657-15674
START_RANGE client8x socket892 180 15674-15692
START_RANGE client8y socket893 180 15692-15710
START_RANGE client8z socket894 170 15710-15727
START_RANGE client8aa socket895 180 15727-15745
START_RANGE client8ab socket896 170 15745-15762

START_RANGE client9a socket897 180 15762-15780
START_RANGE client9b socket898 180 15780-15798
START_RANGE client9c socket899 170 15798-15815
START_RANGE client9d socket900 180 15815-15833
START_RANGE client9e socket901 170 15833-15850
START_RANGE client9f socket902 180 15850-15868
START_RANGE client9g socket903 180 15868-15886
START_RANGE client9h socket904 170 15886-15903
START_RANGE client9i socket905 180 15903-15921
START_RANGE client9j socket906 170 15921-15938
START_RANGE client9k socket907 180 15938-15956
START_RANGE client9l socket908 180 15956-15974
START_RANGE client9m socket909 170 15974-15991
START_RANGE client9n socket910 180 15991-16009
START_RANGE client9a socket911 170 16009-16026
START_RANGE client9b socket912 180 16026-16044
START_RANGE client9c socket913 180 16044-16062
START_RANGE client9d socket914 170 16062-16079
START_RANGE client9e socket915 180 16079-16097
START_RANGE client9f socket916 170 16097-16114
START_RANGE client9g socket917 180 16114-16132
START_RANGE client9h socket918 170 16132-16149
START_RANGE client9i socket919 180 16149-16167
START_RANGE client9j socket920 180 16167-16185
START_RANGE client9k socket921 170 16185-16202
START_RANGE client9l socket922 180 16202-16220

```
START_RANGE client9m socket923 170 16220-16237
START_RANGE client9n socket924 180 16237-16255
START_RANGE client9a socket925 180 16255-16273
START_RANGE client9b socket926 170 16273-16290
START_RANGE client9c socket927 180 16290-16308
START_RANGE client9d socket928 170 16308-16325
START_RANGE client9e socket929 180 16325-16343
START_RANGE client9f socket930 180 16343-16361
START_RANGE client9g socket931 170 16361-16378
START_RANGE client9h socket932 180 16378-16396
START_RANGE client9i socket933 170 16396-16413
START_RANGE client9j socket934 180 16413-16431
START_RANGE client9k socket935 180 16431-16449
START_RANGE client9l socket936 170 16449-16466
START_RANGE client9m socket937 180 16466-16484
START_RANGE client9n socket938 170 16484-16501
START_RANGE client9a socket939 180 16501-16519
START_RANGE client9b socket940 180 16519-16537
START_RANGE client9c socket941 170 16537-16554
START_RANGE client9d socket942 180 16554-16572
START_RANGE client9e socket943 170 16572-16589
START_RANGE client9f socket944 180 16589-16607
START_RANGE client9g socket945 170 16607-16624
START_RANGE client9h socket946 180 16624-16642
START_RANGE client9i socket947 180 16642-16660
START_RANGE client9j socket948 170 16660-16677
START_RANGE client9k socket949 180 16677-16695
START_RANGE client9l socket950 170 16695-16712
START_RANGE client9m socket951 180 16712-16730
START_RANGE client9n socket952 180 16730-16748

START_RANGE client9o socket953 170 16748-16765
START_RANGE client9p socket954 180 16765-16783
START_RANGE client9q socket955 170 16783-16800
START_RANGE client9r socket956 180 16800-16818
START_RANGE client9s socket957 180 16818-16836
START_RANGE client9t socket958 170 16836-16853
START_RANGE client9u socket959 180 16853-16871
START_RANGE client9v socket960 170 16871-16888
START_RANGE client9w socket961 180 16888-16906
START_RANGE client9x socket962 180 16906-16924
START_RANGE client9y socket963 170 16924-16941
START_RANGE client9z socket964 180 16941-16959
START_RANGE client9aa socket965 170 16959-16976
START_RANGE client9ab socket966 180 16976-16994
START_RANGE client9o socket967 180 16994-17012
START_RANGE client9p socket968 170 17012-17029
START_RANGE client9q socket969 180 17029-17047
START_RANGE client9r socket970 170 17047-17064
START_RANGE client9s socket971 180 17064-17082
START_RANGE client9t socket972 170 17082-17099
START_RANGE client9u socket973 180 17099-17117
START_RANGE client9v socket974 180 17117-17135
START_RANGE client9w socket975 170 17135-17152
START_RANGE client9x socket976 180 17152-17170
START_RANGE client9y socket977 170 17170-17187
START_RANGE client9z socket978 180 17187-17205
START_RANGE client9aa socket979 180 17205-17223
START_RANGE client9ab socket980 170 17223-17240
START_RANGE client9o socket981 180 17240-17258
START_RANGE client9p socket982 170 17258-17275
START_RANGE client9q socket983 180 17275-17293
START_RANGE client9r socket984 180 17293-17311
START_RANGE client9s socket985 170 17311-17328
START_RANGE client9t socket986 180 17328-17346
START_RANGE client9u socket987 170 17346-17363
START_RANGE client9v socket988 180 17363-17381
START_RANGE client9w socket989 180 17381-17399
START_RANGE client9x socket990 170 17399-17416
START_RANGE client9y socket991 180 17416-17434
START_RANGE client9z socket992 170 17434-17451
START_RANGE client9aa socket993 180 17451-17469
START_RANGE client9ab socket994 180 17469-17487
START_RANGE client9o socket995 170 17487-17504
START_RANGE client9p socket996 180 17504-17522
START_RANGE client9q socket997 170 17522-17539
START_RANGE client9r socket998 180 17539-17557
START_RANGE client9s socket999 170 17557-17574
START_RANGE client9t socket1000 180 17574-17592
START_RANGE client9u socket1001 180 17592-17610
START_RANGE client9v socket1002 170 17610-17627
START_RANGE client9w socket1003 180 17627-17645
START_RANGE client9x socket1004 170 17645-17662
START_RANGE client9y socket1005 180 17662-17680
START_RANGE client9z socket1006 180 17680-17698
START_RANGE client9aa socket1007 170 17698-17715
START_RANGE client9ab socket1008 180 17715-17733

#elif MASTER_NUM4
START_RANGE client10a socket1009 170 17733-17750
START_RANGE client10b socket1010 180 17750-17768
START_RANGE client10c socket1011 180 17768-17786
START_RANGE client10d socket1012 170 17786-17803
START_RANGE client10e socket1013 180 17803-17821
START_RANGE client10f socket1014 170 17821-17838
START_RANGE client10g socket1015 180 17838-17856
START_RANGE client10h socket1016 180 17856-17874
START_RANGE client10i socket1017 170 17874-17891
START_RANGE client10j socket1018 180 17891-17909
START_RANGE client10k socket1019 170 17909-17926
START_RANGE client10l socket1020 180 17926-17944
START_RANGE client10m socket1021 180 17944-17962
START_RANGE client10n socket1022 170 17962-17979
START_RANGE client10a socket1023 180 17979-17997
START_RANGE client10b socket1024 170 17997-18014
START_RANGE client10c socket1025 180 18014-18032
START_RANGE client10d socket1026 170 18032-18049
START_RANGE client10e socket1027 180 18049-18067
START_RANGE client10f socket1028 180 18067-18085
START_RANGE client10g socket1029 170 18085-18102
START_RANGE client10h socket1030 180 18102-18120
START_RANGE client10i socket1031 170 18120-18137
START_RANGE client10j socket1032 180 18137-18155
START_RANGE client10k socket1033 180 18155-18173
START_RANGE client10l socket1034 170 18173-18190
START_RANGE client10m socket1035 180 18190-18208
START_RANGE client10n socket1036 170 18208-18225
START_RANGE client10a socket1037 180 18225-18243
START_RANGE client10b socket1038 180 18243-18261
START_RANGE client10c socket1039 170 18261-18278
START_RANGE client10d socket1040 180 18278-18296
START_RANGE client10e socket1041 170 18296-18313
START_RANGE client10f socket1042 180 18313-18331
START_RANGE client10g socket1043 180 18331-18349
START_RANGE client10h socket1044 170 18349-18366
START_RANGE client10i socket1045 180 18366-18384
START_RANGE client10j socket1046 170 18384-18401
START_RANGE client10k socket1047 180 18401-18419
START_RANGE client10l socket1048 180 18419-18437
START_RANGE client10m socket1049 170 18437-18454
START_RANGE client10n socket1050 180 18454-18472
START_RANGE client10a socket1051 170 18472-18489
START_RANGE client10b socket1052 180 18489-18507
START_RANGE client10c socket1053 170 18507-18524
START_RANGE client10d socket1054 180 18524-18542
START_RANGE client10e socket1055 180 18542-18560
START_RANGE client10f socket1056 170 18560-18577
START_RANGE client10g socket1057 180 18577-18595
START_RANGE client10h socket1058 170 18595-18612
START_RANGE client10i socket1059 180 18612-18630
START_RANGE client10j socket1060 180 18630-18648
START_RANGE client10k socket1061 170 18648-18665
START_RANGE client10l socket1062 180 18665-18683
START_RANGE client10m socket1063 170 18683-18700
START_RANGE client10n socket1064 180 18700-18718

START_RANGE client10o socket1065 180 18718-18736
START_RANGE client10p socket1066 170 18736-18753
START_RANGE client10q socket1067 180 18753-18771
START_RANGE client10r socket1068 170 18771-18788
START_RANGE client10s socket1069 180 18788-18806
START_RANGE client10t socket1070 180 18806-18824
START_RANGE client10u socket1071 170 18824-18841
START_RANGE client10v socket1072 180 18841-18859
START_RANGE client10w socket1073 170 18859-18876
START_RANGE client10x socket1074 180 18876-18894
START_RANGE client10y socket1075 180 18894-18912
START_RANGE client10z socket1076 170 18912-18929
START_RANGE client10aa socket1077 180 18929-18947
START_RANGE client10ab socket1078 170 18947-18964
START_RANGE client10o socket1079 180 18964-18982
START_RANGE client10p socket1080 170 18982-18999
START_RANGE client10q socket1081 180 18999-19017
START_RANGE client10r socket1082 180 19017-19035
START_RANGE client10s socket1083 170 19035-19052
START_RANGE client10t socket1084 180 19052-19070
START_RANGE client10u socket1085 170 19070-19087
START_RANGE client10v socket1086 180 19087-19105
START_RANGE client10w socket1087 180 19105-19123
START_RANGE client10x socket1088 170 19123-19140
START_RANGE client10y socket1089 180 19140-19158
START_RANGE client10z socket1090 170 19158-19175
START_RANGE client10aa socket1091 180 19175-19193
START_RANGE client10ab socket1092 180 19193-19211
START_RANGE client10o socket1093 170 19211-19228
START_RANGE client10p socket1094 180 19228-19246
START_RANGE client10q socket1095 170 19246-19263
START_RANGE client10r socket1096 180 19263-19281
START_RANGE client10s socket1097 180 19281-19299
START_RANGE client10t socket1098 170 19299-19316
START_RANGE client10u socket1099 180 19316-19334
START_RANGE client10v socket1100 170 19334-19351
START_RANGE client10w socket1101 180 19351-19369
START_RANGE client10x socket1102 180 19369-19387
START_RANGE client10y socket1103 170 19387-19404
START_RANGE client10z socket1104 180 19404-19422
START_RANGE client10aa socket1105 170 19422-19439
START_RANGE client10ab socket1106 180 19439-19457
START_RANGE client10o socket1107 170 19457-19474
START_RANGE client10p socket1108 180 19474-19492
START_RANGE client10q socket1109 180 19492-19510
START_RANGE client10r socket1110 170 19510-19527
START_RANGE client10s socket1111 180 19527-19545
START_RANGE client10t socket1112 170 19545-19562
START_RANGE client10u socket1113 180 19562-19580
START_RANGE client10v socket1114 180 19580-19598
START_RANGE client10w socket1115 170 19598-19615
START_RANGE client10x socket1116 180 19615-19633
```

START_RANGE client10y socket1117 170 19633-19650
START_RANGE client10z socket1118 180 19650-19668
START_RANGE client10aa socket1119 180 19668-19686
START_RANGE client10ab socket1120 170 19686-19703

START_RANGE client11a socket1121 180 19703-19721
START_RANGE client11b socket1122 170 19721-19738
START_RANGE client11c socket1123 180 19738-19756
START_RANGE client11d socket1124 180 19756-19774
START_RANGE client11e socket1125 170 19774-19791
START_RANGE client11f socket1126 180 19791-19809
START_RANGE client11g socket1127 170 19809-19826
START_RANGE client11h socket1128 180 19826-19844
START_RANGE client11i socket1129 180 19844-19862
START_RANGE client11j socket1130 170 19862-19879
START_RANGE client11k socket1131 180 19879-19897
START_RANGE client11l socket1132 170 19897-19914
START_RANGE client11m socket1133 180 19914-19932
START_RANGE client11n socket1134 170 19932-19949
START_RANGE client11a socket1135 180 19949-19967
START_RANGE client11b socket1136 180 19967-19985
START_RANGE client11c socket1137 170 19985-20002
START_RANGE client11d socket1138 180 20002-20020
START_RANGE client11e socket1139 170 20020-20037
START_RANGE client11f socket1140 180 20037-20055
START_RANGE client11g socket1141 180 20055-20073
START_RANGE client11h socket1142 170 20073-20090
START_RANGE client11i socket1143 180 20090-20108
START_RANGE client11j socket1144 170 20108-20125
START_RANGE client11k socket1145 180 20125-20143
START_RANGE client11l socket1146 180 20143-20161
START_RANGE client11m socket1147 170 20161-20178
START_RANGE client11n socket1148 180 20178-20196
START_RANGE client11a socket1149 170 20196-20213
START_RANGE client11b socket1150 180 20213-20231
START_RANGE client11c socket1151 180 20231-20249
START_RANGE client11d socket1152 170 20249-20266
START_RANGE client11e socket1153 180 20266-20284
START_RANGE client11f socket1154 170 20284-20301
START_RANGE client11g socket1155 180 20301-20319
START_RANGE client11h socket1156 180 20319-20337
START_RANGE client11i socket1157 170 20337-20354
START_RANGE client11j socket1158 180 20354-20372
START_RANGE client11k socket1159 170 20372-20389
START_RANGE client11l socket1160 180 20389-20407
START_RANGE client11m socket1161 170 20407-20424
START_RANGE client11n socket1162 180 20424-20442
START_RANGE client11a socket1163 180 20442-20460
START_RANGE client11b socket1164 170 20460-20477
START_RANGE client11c socket1165 180 20477-20495
START_RANGE client11d socket1166 170 20495-20512
START_RANGE client11e socket1167 180 20512-20530
START_RANGE client11f socket1168 180 20530-20548
START_RANGE client11g socket1169 170 20548-20565
START_RANGE client11h socket1170 180 20565-20583
START_RANGE client11i socket1171 170 20583-20600
START_RANGE client11j socket1172 180 20600-20618
START_RANGE client11k socket1173 180 20618-20636
START_RANGE client11l socket1174 170 20636-20653
START_RANGE client11m socket1175 180 20653-20671
START_RANGE client11n socket1176 170 20671-20688

START_RANGE client11o socket1177 180 20688-20706
START_RANGE client11p socket1178 180 20706-20724
START_RANGE client11q socket1179 170 20724-20741
START_RANGE client11r socket1180 180 20741-20759
START_RANGE client11s socket1181 170 20759-20776
START_RANGE client11t socket1182 180 20776-20794
START_RANGE client11u socket1183 180 20794-20812
START_RANGE client11v socket1184 170 20812-20829
START_RANGE client11w socket1185 180 20829-20847
START_RANGE client11x socket1186 170 20847-20864
START_RANGE client11y socket1187 180 20864-20882
START_RANGE client11z socket1188 170 20882-20899
START_RANGE client11aa socket1189 180 20899-20917
START_RANGE client11ab socket1190 180 20917-20935
START_RANGE client11o socket1191 170 20935-20952
START_RANGE client11p socket1192 180 20952-20970
START_RANGE client11q socket1193 170 20970-20987
START_RANGE client11r socket1194 180 20987-21005
START_RANGE client11s socket1195 180 21005-21023
START_RANGE client11t socket1196 170 21023-21040
START_RANGE client11u socket1197 180 21040-21058
START_RANGE client11v socket1198 170 21058-21075
START_RANGE client11w socket1199 180 21075-21093
START_RANGE client11x socket1200 180 21093-21111
START_RANGE client11y socket1201 170 21111-21128
START_RANGE client11z socket1202 180 21128-21146
START_RANGE client11aa socket1203 170 21146-21163
START_RANGE client11ab socket1204 180 21163-21181
START_RANGE client11o socket1205 180 21181-21199
START_RANGE client11p socket1206 170 21199-21216
START_RANGE client11q socket1207 180 21216-21234
START_RANGE client11r socket1208 170 21234-21251
START_RANGE client11s socket1209 180 21251-21269
START_RANGE client11t socket1210 180 21269-21287
START_RANGE client11u socket1211 170 21287-21304
START_RANGE client11v socket1212 180 21304-21322
START_RANGE client11w socket1213 170 21322-21339
START_RANGE client11x socket1214 180 21339-21357
START_RANGE client11y socket1215 170 21357-21374
START_RANGE client11z socket1216 180 21374-21392
START_RANGE client11aa socket1217 180 21392-21410
START_RANGE client11ab socket1218 170 21410-21427
START_RANGE client11o socket1219 180 21427-21445
START_RANGE client11p socket1220 170 21445-21462
START_RANGE client11q socket1221 180 21462-21480
START_RANGE client11r socket1222 180 21480-21498
START_RANGE client11s socket1223 170 21498-21515
START_RANGE client11t socket1224 180 21515-21533
START_RANGE client11u socket1225 170 21533-21550
START_RANGE client11v socket1226 180 21550-21568
START_RANGE client11w socket1227 180 21568-21586
START_RANGE client11x socket1228 170 21586-21603
START_RANGE client11y socket1229 180 21603-21621
START_RANGE client11z socket1230 170 21621-21638
START_RANGE client11aa socket1231 180 21638-21656
START_RANGE client11ab socket1232 180 21656-21674

START_RANGE client12a socket1233 170 21674-21691
START_RANGE client12b socket1234 180 21691-21709
START_RANGE client12c socket1235 170 21709-21726
START_RANGE client12d socket1236 180 21726-21744
START_RANGE client12e socket1237 180 21744-21762
START_RANGE client12f socket1238 170 21762-21779
START_RANGE client12g socket1239 180 21779-21797
START_RANGE client12h socket1240 170 21797-21814
START_RANGE client12i socket1241 180 21814-21832
START_RANGE client12j socket1242 170 21832-21849
START_RANGE client12k socket1243 180 21849-21867
START_RANGE client12l socket1244 180 21867-21885
START_RANGE client12m socket1245 170 21885-21902
START_RANGE client12n socket1246 180 21902-21920
START_RANGE client12a socket1247 170 21920-21937
START_RANGE client12b socket1248 180 21937-21955
START_RANGE client12c socket1249 180 21955-21973
START_RANGE client12d socket1250 170 21973-21990
START_RANGE client12e socket1251 180 21990-22008
START_RANGE client12f socket1252 170 22008-22025
START_RANGE client12g socket1253 180 22025-22043
START_RANGE client12h socket1254 180 22043-22061
START_RANGE client12i socket1255 170 22061-22078
START_RANGE client12j socket1256 180 22078-22096
START_RANGE client12k socket1257 170 22096-22113
START_RANGE client12l socket1258 180 22113-22131
START_RANGE client12m socket1259 180 22131-22149
START_RANGE client12n socket1260 170 22149-22166
START_RANGE client12a socket1261 180 22166-22184
START_RANGE client12b socket1262 170 22184-22201
START_RANGE client12c socket1263 180 22201-22219
START_RANGE client12d socket1264 180 22219-22237
START_RANGE client12e socket1265 170 22237-22254
START_RANGE client12f socket1266 180 22254-22272
START_RANGE client12g socket1267 170 22272-22289
START_RANGE client12h socket1268 180 22289-22307
START_RANGE client12i socket1269 170 22307-22324
START_RANGE client12j socket1270 180 22324-22342
START_RANGE client12k socket1271 180 22342-22360
START_RANGE client12l socket1272 170 22360-22377
START_RANGE client12m socket1273 180 22377-22395
START_RANGE client12n socket1274 170 22395-22412
START_RANGE client12a socket1275 180 22412-22430
START_RANGE client12b socket1276 180 22430-22448
START_RANGE client12c socket1277 170 22448-22465
START_RANGE client12d socket1278 180 22465-22483
START_RANGE client12e socket1279 170 22483-22500
START_RANGE client12f socket1280 180 22500-22518
START_RANGE client12g socket1281 180 22518-22536
START_RANGE client12h socket1282 170 22536-22553
START_RANGE client12i socket1283 180 22553-22571
START_RANGE client12j socket1284 170 22571-22588
START_RANGE client12k socket1285 180 22588-22606
START_RANGE client12l socket1286 180 22606-22624
START_RANGE client12m socket1287 170 22624-22641
START_RANGE client12n socket1288 180 22641-22659

START_RANGE client12o socket1289 170 22659-22676
START_RANGE client12p socket1290 180 22676-22694
START_RANGE client12q socket1291 180 22694-22712
START_RANGE client12r socket1292 170 22712-22729
START_RANGE client12s socket1293 180 22729-22747
START_RANGE client12t socket1294 170 22747-22764
START_RANGE client12u socket1295 180 22764-22782
START_RANGE client12v socket1296 170 22782-22799
START_RANGE client12w socket1297 180 22799-22817
START_RANGE client12x socket1298 180 22817-22835
START_RANGE client12y socket1299 170 22835-22852
START_RANGE client12z socket1300 180 22852-22870
START_RANGE client12aa socket1301 170 22870-22887
START_RANGE client12ab socket1302 180 22887-22905
START_RANGE client12o socket1303 180 22905-22923
START_RANGE client12p socket1304 170 22923-22940
START_RANGE client12q socket1305 180 22940-22958
START_RANGE client12r socket1306 170 22958-22975
START_RANGE client12s socket1307 180 22975-22993
START_RANGE client12t socket1308 180 22993-23011
START_RANGE client12u socket1309 170 23011-23028
START_RANGE client12v socket1310 180 23028-23046

```
START_RANGE client12w socket1311 170 23046-23063
START_RANGE client12x socket1312 180 23063-23081
START_RANGE client12y socket1313 180 23081-23099
START_RANGE client12z socket1314 170 23099-23116
START_RANGE client12aa socket1315 180 23116-23134
START_RANGE client12ab socket1316 170 23134-23151
START_RANGE client12o socket1317 180 23151-23169
START_RANGE client12p socket1318 180 23169-23187
START_RANGE client12q socket1319 170 23187-23204
START_RANGE client12r socket1320 180 23204-23222
START_RANGE client12s socket1321 170 23222-23239
START_RANGE client12t socket1322 180 23239-23257
START_RANGE client12u socket1323 170 23257-23274
START_RANGE client12v socket1324 180 23274-23292
START_RANGE client12w socket1325 180 23292-23310
START_RANGE client12x socket1326 170 23310-23327
START_RANGE client12y socket1327 180 23327-23345
START_RANGE client12z socket1328 170 23345-23362
START_RANGE client12aa socket1329 180 23362-23380
START_RANGE client12ab socket1330 180 23380-23398
START_RANGE client12o socket1331 170 23398-23415
START_RANGE client12p socket1332 180 23415-23433
START_RANGE client12q socket1333 170 23433-23450
START_RANGE client12r socket1334 180 23450-23468
START_RANGE client12s socket1335 180 23468-23486
START_RANGE client12t socket1336 170 23486-23503
START_RANGE client12u socket1337 180 23503-23521
START_RANGE client12v socket1338 170 23521-23538
START_RANGE client12w socket1339 180 23538-23556
START_RANGE client12x socket1340 180 23556-23574
START_RANGE client12y socket1341 170 23574-23591
START_RANGE client12z socket1342 180 23591-23609
START_RANGE client12aa socket1343 170 23609-23626
START_RANGE client12ab socket1344 180 23626-23644

#elif MASTER_NUM5
START_RANGE client13a socket1345 180 23644-23662
START_RANGE client13b socket1346 170 23662-23679
START_RANGE client13c socket1347 180 23679-23697
START_RANGE client13d socket1348 170 23697-23714
START_RANGE client13e socket1349 180 23714-23732
START_RANGE client13f socket1350 170 23732-23749
START_RANGE client13g socket1351 180 23749-23767
START_RANGE client13h socket1352 180 23767-23785
START_RANGE client13i socket1353 170 23785-23802
START_RANGE client13j socket1354 180 23802-23820
START_RANGE client13k socket1355 170 23820-23837
START_RANGE client13l socket1356 180 23837-23855
START_RANGE client13m socket1357 180 23855-23873
START_RANGE client13n socket1358 170 23873-23890
START_RANGE client13a socket1359 180 23890-23908
START_RANGE client13b socket1360 170 23908-23925
START_RANGE client13c socket1361 180 23925-23943
START_RANGE client13d socket1362 180 23943-23961
START_RANGE client13e socket1363 170 23961-23978
START_RANGE client13f socket1364 180 23978-23996
START_RANGE client13g socket1365 170 23996-24013
START_RANGE client13h socket1366 180 24013-24031
START_RANGE client13i socket1367 180 24031-24049
START_RANGE client13j socket1368 170 24049-24066
START_RANGE client13k socket1369 180 24066-24084
START_RANGE client13l socket1370 170 24084-24101
START_RANGE client13m socket1371 180 24101-24119
START_RANGE client13n socket1372 180 24119-24137
START_RANGE client13a socket1373 170 24137-24154
START_RANGE client13b socket1374 180 24154-24172
START_RANGE client13c socket1375 170 24172-24189
START_RANGE client13d socket1376 180 24189-24207
START_RANGE client13e socket1377 170 24207-24224
START_RANGE client13f socket1378 180 24224-24242
START_RANGE client13g socket1379 180 24242-24260
START_RANGE client13h socket1380 170 24260-24277
START_RANGE client13i socket1381 180 24277-24295
START_RANGE client13j socket1382 170 24295-24312
START_RANGE client13k socket1383 180 24312-24330
START_RANGE client13l socket1384 180 24330-24348
START_RANGE client13m socket1385 170 24348-24365
START_RANGE client13n socket1386 180 24365-24383
START_RANGE client13a socket1387 170 24383-24400
START_RANGE client13b socket1388 180 24400-24418
START_RANGE client13c socket1389 180 24418-24436
START_RANGE client13d socket1390 170 24436-24453
START_RANGE client13e socket1391 180 24453-24471
START_RANGE client13f socket1392 170 24471-24488
START_RANGE client13g socket1393 180 24488-24506
START_RANGE client13h socket1394 180 24506-24524
START_RANGE client13i socket1395 170 24524-24541
START_RANGE client13j socket1396 180 24541-24559
START_RANGE client13k socket1397 170 24559-24576
START_RANGE client13l socket1398 180 24576-24594
START_RANGE client13m socket1399 180 24594-24612
START_RANGE client13n socket1400 170 24612-24629

START_RANGE client13o socket1401 180 24629-24647
START_RANGE client13p socket1402 170 24647-24664
START_RANGE client13q socket1403 180 24664-24682
START_RANGE client13r socket1404 170 24682-24699
START_RANGE client13s socket1405 180 24699-24717
START_RANGE client13t socket1406 180 24717-24735
START_RANGE client13u socket1407 170 24735-24752
START_RANGE client13v socket1408 180 24752-24770
START_RANGE client13w socket1409 170 24770-24787
START_RANGE client13x socket1410 180 24787-24805
START_RANGE client13y socket1411 180 24805-24823
START_RANGE client13z socket1412 170 24823-24840
START_RANGE client13aa socket1413 180 24840-24858
START_RANGE client13ab socket1414 170 24858-24875
START_RANGE client13o socket1415 180 24875-24893
START_RANGE client13p socket1416 180 24893-24911
START_RANGE client13q socket1417 170 24911-24928
START_RANGE client13r socket1418 180 24928-24946
START_RANGE client13s socket1419 170 24946-24963
START_RANGE client13t socket1420 180 24963-24981
START_RANGE client13u socket1421 180 24981-24999
START_RANGE client13v socket1422 170 24999-25016
START_RANGE client13w socket1423 180 25016-25034
START_RANGE client13x socket1424 170 25034-25051
START_RANGE client13y socket1425 180 25051-25069
START_RANGE client13z socket1426 180 25069-25087
START_RANGE client13aa socket1427 170 25087-25104
START_RANGE client13ab socket1428 180 25104-25122
START_RANGE client13o socket1429 170 25122-25139
START_RANGE client13p socket1430 180 25139-25157
START_RANGE client13q socket1431 170 25157-25174
START_RANGE client13r socket1432 180 25174-25192
START_RANGE client13s socket1433 180 25192-25210
START_RANGE client13t socket1434 170 25210-25227
START_RANGE client13u socket1435 180 25227-25245
START_RANGE client13v socket1436 170 25245-25262
START_RANGE client13w socket1437 180 25262-25280
START_RANGE client13x socket1438 180 25280-25298
START_RANGE client13y socket1439 170 25298-25315
START_RANGE client13z socket1440 180 25315-25333
START_RANGE client13aa socket1441 170 25333-25350
START_RANGE client13ab socket1442 180 25350-25368
START_RANGE client13o socket1443 180 25368-25386
START_RANGE client13p socket1444 170 25386-25403
START_RANGE client13q socket1445 180 25403-25421
START_RANGE client13r socket1446 170 25421-25438
START_RANGE client13s socket1447 180 25438-25456
START_RANGE client13t socket1448 180 25456-25474
START_RANGE client13u socket1449 170 25474-25491
START_RANGE client13v socket1450 180 25491-25509
START_RANGE client13w socket1451 170 25509-25526
START_RANGE client13x socket1452 180 25526-25544
START_RANGE client13y socket1453 180 25544-25562
START_RANGE client13z socket1454 170 25562-25579
START_RANGE client13aa socket1455 180 25579-25597
START_RANGE client13ab socket1456 170 25597-25614

START_RANGE client14a socket1457 180 25614-25632
START_RANGE client14b socket1458 170 25632-25649
START_RANGE client14c socket1459 180 25649-25667
START_RANGE client14d socket1460 180 25667-25685
START_RANGE client14e socket1461 170 25685-25702
START_RANGE client14f socket1462 180 25702-25720
START_RANGE client14g socket1463 170 25720-25737
START_RANGE client14h socket1464 180 25737-25755
START_RANGE client14i socket1465 180 25755-25773
START_RANGE client14j socket1466 170 25773-25790
START_RANGE client14k socket1467 180 25790-25808
START_RANGE client14l socket1468 170 25808-25825
START_RANGE client14m socket1469 180 25825-25843
START_RANGE client14n socket1470 180 25843-25861
START_RANGE client14a socket1471 170 25861-25878
START_RANGE client14b socket1472 180 25878-25896
START_RANGE client14c socket1473 170 25896-25913
START_RANGE client14d socket1474 180 25913-25931
START_RANGE client14e socket1475 180 25931-25949
START_RANGE client14f socket1476 170 25949-25966
START_RANGE client14g socket1477 180 25966-25984
START_RANGE client14h socket1478 170 25984-26001
START_RANGE client14i socket1479 180 26001-26019
START_RANGE client14j socket1480 180 26019-26037
START_RANGE client14k socket1481 170 26037-26054
START_RANGE client14l socket1482 180 26054-26072
START_RANGE client14m socket1483 170 26072-26089
START_RANGE client14n socket1484 180 26089-26107
START_RANGE client14a socket1485 170 26107-26124
START_RANGE client14b socket1486 180 26124-26142
START_RANGE client14c socket1487 180 26142-26160
START_RANGE client14d socket1488 170 26160-26177
START_RANGE client14e socket1489 180 26177-26195
START_RANGE client14f socket1490 170 26195-26212
START_RANGE client14g socket1491 180 26212-26230
START_RANGE client14h socket1492 180 26230-26248
START_RANGE client14i socket1493 170 26248-26265
START_RANGE client14j socket1494 180 26265-26283
START_RANGE client14k socket1495 170 26283-26300
START_RANGE client14l socket1496 180 26300-26318
START_RANGE client14m socket1497 180 26318-26336
START_RANGE client14n socket1498 170 26336-26353
START_RANGE client14a socket1499 180 26353-26371
START_RANGE client14b socket1500 170 26371-26388
START_RANGE client14c socket1501 180 26388-26406
START_RANGE client14d socket1502 180 26406-26424
START_RANGE client14e socket1503 170 26424-26441
START_RANGE client14f socket1504 180 26441-26459
```

START_RANGE client14g socket1505 170 26459-26476
START_RANGE client14h socket1506 180 26476-26494
START_RANGE client14i socket1507 180 26494-26512
START_RANGE client14j socket1508 170 26512-26529
START_RANGE client14k socket1509 180 26529-26547
START_RANGE client14l socket1510 170 26547-26564
START_RANGE client14m socket1511 180 26564-26582
START_RANGE client14n socket1512 170 26582-26599

START_RANGE client14o socket1513 180 26599-26617
START_RANGE client14p socket1514 180 26617-26635
START_RANGE client14q socket1515 170 26635-26652
START_RANGE client14r socket1516 180 26652-26670
START_RANGE client14s socket1517 170 26670-26687
START_RANGE client14t socket1518 180 26687-26705
START_RANGE client14u socket1519 180 26705-26723
START_RANGE client14v socket1520 170 26723-26740
START_RANGE client14w socket1521 180 26740-26758
START_RANGE client14x socket1522 170 26758-26775
START_RANGE client14y socket1523 180 26775-26793
START_RANGE client14z socket1524 180 26793-26811
START_RANGE client14aa socket1525 170 26811-26828
START_RANGE client14ab socket1526 180 26828-26846
START_RANGE client14o socket1527 170 26846-26863
START_RANGE client14p socket1528 180 26863-26881
START_RANGE client14q socket1529 180 26881-26899
START_RANGE client14r socket1530 170 26899-26916
START_RANGE client14s socket1531 180 26916-26934
START_RANGE client14t socket1532 170 26934-26951
START_RANGE client14u socket1533 180 26951-26969
START_RANGE client14v socket1534 180 26969-26987
START_RANGE client14w socket1535 170 26987-27004
START_RANGE client14x socket1536 180 27004-27022
START_RANGE client14y socket1537 170 27022-27039
START_RANGE client14z socket1538 180 27039-27057
START_RANGE client14aa socket1539 170 27057-27074
START_RANGE client14ab socket1540 180 27074-27092
START_RANGE client14o socket1541 180 27092-27110
START_RANGE client14p socket1542 170 27110-27127
START_RANGE client14q socket1543 180 27127-27145
START_RANGE client14r socket1544 170 27145-27162
START_RANGE client14s socket1545 180 27162-27180
START_RANGE client14t socket1546 180 27180-27198
START_RANGE client14u socket1547 170 27198-27215
START_RANGE client14v socket1548 180 27215-27233
START_RANGE client14w socket1549 170 27233-27250
START_RANGE client14x socket1550 180 27250-27268
START_RANGE client14y socket1551 180 27268-27286
START_RANGE client14z socket1552 170 27286-27303
START_RANGE client14aa socket1553 180 27303-27321
START_RANGE client14ab socket1554 170 27321-27338
START_RANGE client14o socket1555 180 27338-27356
START_RANGE client14p socket1556 180 27356-27374
START_RANGE client14q socket1557 170 27374-27391
START_RANGE client14r socket1558 180 27391-27409
START_RANGE client14s socket1559 170 27409-27426
START_RANGE client14t socket1560 180 27426-27444
START_RANGE client14u socket1561 180 27444-27462
START_RANGE client14v socket1562 170 27462-27479
START_RANGE client14w socket1563 180 27479-27497
START_RANGE client14x socket1564 170 27497-27514
START_RANGE client14y socket1565 180 27514-27532
START_RANGE client14z socket1566 170 27532-27549
START_RANGE client14aa socket1567 180 27549-27567
START_RANGE client14ab socket1568 180 27567-27585

START_RANGE client15a socket1569 170 27585-27602
START_RANGE client15b socket1570 180 27602-27620
START_RANGE client15c socket1571 170 27620-27637
START_RANGE client15d socket1572 180 27637-27655
START_RANGE client15e socket1573 180 27655-27673
START_RANGE client15f socket1574 170 27673-27690
START_RANGE client15g socket1575 180 27690-27708
START_RANGE client15h socket1576 170 27708-27725
START_RANGE client15i socket1577 180 27725-27743
START_RANGE client15j socket1578 180 27743-27761
START_RANGE client15k socket1579 170 27761-27778
START_RANGE client15l socket1580 180 27778-27796
START_RANGE client15m socket1581 170 27796-27813
START_RANGE client15n socket1582 180 27813-27831
START_RANGE client15a socket1583 180 27831-27849
START_RANGE client15b socket1584 170 27849-27866
START_RANGE client15c socket1585 180 27866-27884
START_RANGE client15d socket1586 170 27884-27901
START_RANGE client15e socket1587 180 27901-27919
START_RANGE client15f socket1588 180 27919-27937
START_RANGE client15g socket1589 170 27937-27954
START_RANGE client15h socket1590 180 27954-27972
START_RANGE client15i socket1591 170 27972-27989
START_RANGE client15j socket1592 180 27989-28007
START_RANGE client15k socket1593 170 28007-28024
START_RANGE client15l socket1594 180 28024-28042
START_RANGE client15m socket1595 180 28042-28060
START_RANGE client15n socket1596 170 28060-28077
START_RANGE client15a socket1597 180 28077-28095
START_RANGE client15b socket1598 170 28095-28112
START_RANGE client15c socket1599 180 28112-28130
START_RANGE client15d socket1600 180 28130-28148
START_RANGE client15e socket1601 170 28148-28165
START_RANGE client15f socket1602 180 28165-28183
START_RANGE client15g socket1603 170 28183-28200
START_RANGE client15h socket1604 180 28200-28218
START_RANGE client15i socket1605 180 28218-28236
START_RANGE client15j socket1606 170 28236-28253
START_RANGE client15k socket1607 180 28253-28271
START_RANGE client15l socket1608 170 28271-28288
START_RANGE client15m socket1609 180 28288-28306
START_RANGE client15n socket1610 180 28306-28324
START_RANGE client15a socket1611 170 28324-28341
START_RANGE client15b socket1612 180 28341-28359
START_RANGE client15c socket1613 170 28359-28376
START_RANGE client15d socket1614 180 28376-28394
START_RANGE client15e socket1615 180 28394-28412
START_RANGE client15f socket1616 170 28412-28429
START_RANGE client15g socket1617 180 28429-28447
START_RANGE client15h socket1618 170 28447-28464
START_RANGE client15i socket1619 180 28464-28482
START_RANGE client15j socket1620 170 28482-28499
START_RANGE client15k socket1621 180 28499-28517
START_RANGE client15l socket1622 180 28517-28535
START_RANGE client15m socket1623 170 28535-28552
START_RANGE client15n socket1624 180 28552-28570

START_RANGE client15o socket1625 170 28570-28587
START_RANGE client15p socket1626 180 28587-28605
START_RANGE client15q socket1627 180 28605-28623
START_RANGE client15r socket1628 170 28623-28640
START_RANGE client15s socket1629 180 28640-28658
START_RANGE client15t socket1630 170 28658-28675
START_RANGE client15u socket1631 180 28675-28693
START_RANGE client15v socket1632 180 28693-28711
START_RANGE client15w socket1633 170 28711-28728
START_RANGE client15x socket1634 180 28728-28746
START_RANGE client15y socket1635 170 28746-28763
START_RANGE client15z socket1636 180 28763-28781
START_RANGE client15aa socket1637 180 28781-28799
START_RANGE client15ab socket1638 170 28799-28816
START_RANGE client15o socket1639 180 28816-28834
START_RANGE client15p socket1640 170 28834-28851
START_RANGE client15q socket1641 180 28851-28869
START_RANGE client15r socket1642 180 28869-28887
START_RANGE client15s socket1643 170 28887-28904
START_RANGE client15t socket1644 180 28904-28922
START_RANGE client15u socket1645 170 28922-28939
START_RANGE client15v socket1646 180 28939-28957
START_RANGE client15w socket1647 170 28957-28974
START_RANGE client15x socket1648 180 28974-28992
START_RANGE client15y socket1649 180 28992-29010
START_RANGE client15z socket1650 170 29010-29027
START_RANGE client15aa socket1651 180 29027-29045
START_RANGE client15ab socket1652 170 29045-29062
START_RANGE client15o socket1653 180 29062-29080
START_RANGE client15p socket1654 180 29080-29098
START_RANGE client15q socket1655 170 29098-29115
START_RANGE client15r socket1656 180 29115-29133
START_RANGE client15s socket1657 170 29133-29150
START_RANGE client15t socket1658 180 29150-29168
START_RANGE client15u socket1659 180 29168-29186
START_RANGE client15v socket1660 170 29186-29203
START_RANGE client15w socket1661 180 29203-29221
START_RANGE client15x socket1662 170 29221-29238
START_RANGE client15y socket1663 180 29238-29256
START_RANGE client15z socket1664 180 29256-29274
START_RANGE client15aa socket1665 170 29274-29291
START_RANGE client15ab socket1666 180 29291-29309
START_RANGE client15o socket1667 170 29309-29326
START_RANGE client15p socket1668 180 29326-29344
START_RANGE client15q socket1669 180 29344-29362
START_RANGE client15r socket1670 170 29362-29379
START_RANGE client15s socket1671 180 29379-29397
START_RANGE client15t socket1672 170 29397-29414
START_RANGE client15u socket1673 180 29414-29432
START_RANGE client15v socket1674 170 29432-29449
START_RANGE client15w socket1675 180 29449-29467
START_RANGE client15x socket1676 180 29467-29485
START_RANGE client15y socket1677 170 29485-29502
START_RANGE client15z socket1678 180 29502-29520
START_RANGE client15aa socket1679 170 29520-29537
START_RANGE client15ab socket1680 180 29537-29555

#elif MASTER_NUM6
START_RANGE client16a socket1681 180 29555-29573
START_RANGE client16b socket1682 170 29573-29590
START_RANGE client16c socket1683 180 29590-29608
START_RANGE client16d socket1684 170 29608-29625
START_RANGE client16e socket1685 180 29625-29643
START_RANGE client16f socket1686 180 29643-29661
START_RANGE client16g socket1687 170 29661-29678
START_RANGE client16h socket1688 180 29678-29696
START_RANGE client16i socket1689 170 29696-29713
START_RANGE client16j socket1690 180 29713-29731
START_RANGE client16k socket1691 180 29731-29749
START_RANGE client16l socket1692 170 29749-29766
START_RANGE client16m socket1693 180 29766-29784
START_RANGE client16n socket1694 170 29784-29801
START_RANGE client16a socket1695 180 29801-29819
START_RANGE client16b socket1696 180 29819-29837
START_RANGE client16c socket1697 170 29837-29854

START_RANGE client16d socket1698 180 29854-29872
START_RANGE client16e socket1699 170 29872-29889
START_RANGE client16f socket1700 180 29889-29907
START_RANGE client16g socket1701 170 29907-29924
START_RANGE client16h socket1702 180 29924-29942
START_RANGE client16i socket1703 180 29942-29960
START_RANGE client16j socket1704 170 29960-29977
START_RANGE client16k socket1705 180 29977-29995
START_RANGE client16l socket1706 170 29995-30012
START_RANGE client16m socket1707 180 30012-30030
START_RANGE client16n socket1708 180 30030-30048
START_RANGE client16a socket1709 170 30048-30065
START_RANGE client16b socket1710 180 30065-30083
START_RANGE client16c socket1711 170 30083-30100
START_RANGE client16d socket1712 180 30100-30118
START_RANGE client16e socket1713 180 30118-30136
START_RANGE client16f socket1714 170 30136-30153
START_RANGE client16g socket1715 180 30153-30171
START_RANGE client16h socket1716 170 30171-30188
START_RANGE client16i socket1717 180 30188-30206
START_RANGE client16j socket1718 180 30206-30224
START_RANGE client16k socket1719 170 30224-30241
START_RANGE client16l socket1720 180 30241-30259
START_RANGE client16m socket1721 170 30259-30276
START_RANGE client16n socket1722 180 30276-30294
START_RANGE client16a socket1723 180 30294-30312
START_RANGE client16b socket1724 170 30312-30329
START_RANGE client16c socket1725 180 30329-30347
START_RANGE client16d socket1726 170 30347-30364
START_RANGE client16e socket1727 180 30364-30382
START_RANGE client16f socket1728 170 30382-30399
START_RANGE client16g socket1729 180 30399-30417
START_RANGE client16h socket1730 180 30417-30435
START_RANGE client16i socket1731 170 30435-30452
START_RANGE client16j socket1732 180 30452-30470
START_RANGE client16k socket1733 170 30470-30487
START_RANGE client16l socket1734 180 30487-30505
START_RANGE client16m socket1735 180 30505-30523
START_RANGE client16n socket1736 170 30523-30540

START_RANGE client16o socket1737 180 30540-30558
START_RANGE client16p socket1738 170 30558-30575
START_RANGE client16q socket1739 180 30575-30593
START_RANGE client16r socket1740 180 30593-30611
START_RANGE client16s socket1741 170 30611-30628
START_RANGE client16t socket1742 180 30628-30646
START_RANGE client16u socket1743 170 30646-30663
START_RANGE client16v socket1744 180 30663-30681
START_RANGE client16w socket1745 180 30681-30699
START_RANGE client16x socket1746 170 30699-30716
START_RANGE client16y socket1747 180 30716-30734
START_RANGE client16z socket1748 170 30734-30751
START_RANGE client16aa socket1749 180 30751-30769
START_RANGE client16ab socket1750 180 30769-30787
START_RANGE client16o socket1751 170 30787-30804
START_RANGE client16p socket1752 180 30804-30822
START_RANGE client16q socket1753 170 30822-30839
START_RANGE client16r socket1754 180 30839-30857
START_RANGE client16s socket1755 170 30857-30874
START_RANGE client16t socket1756 180 30874-30892
START_RANGE client16u socket1757 180 30892-30910
START_RANGE client16v socket1758 170 30910-30927
START_RANGE client16w socket1759 180 30927-30945
START_RANGE client16x socket1760 170 30945-30962
START_RANGE client16y socket1761 180 30962-30980
START_RANGE client16z socket1762 180 30980-30998
START_RANGE client16aa socket1763 170 30998-31015
START_RANGE client16ab socket1764 180 31015-31033
START_RANGE client16o socket1765 170 31033-31050
START_RANGE client16p socket1766 180 31050-31068
START_RANGE client16q socket1767 180 31068-31086
START_RANGE client16r socket1768 170 31086-31103
START_RANGE client16s socket1769 180 31103-31121
START_RANGE client16t socket1770 170 31121-31138
START_RANGE client16u socket1771 180 31138-31156
START_RANGE client16v socket1772 180 31156-31174
START_RANGE client16w socket1773 170 31174-31191
START_RANGE client16x socket1774 180 31191-31209
START_RANGE client16y socket1775 170 31209-31226
START_RANGE client16z socket1776 180 31226-31244
START_RANGE client16aa socket1777 180 31244-31262
START_RANGE client16ab socket1778 170 31262-31279
START_RANGE client16o socket1779 180 31279-31297
START_RANGE client16p socket1780 170 31297-31314
START_RANGE client16q socket1781 180 31314-31332
START_RANGE client16r socket1782 170 31332-31349
START_RANGE client16s socket1783 180 31349-31367
START_RANGE client16t socket1784 180 31367-31385
START_RANGE client16u socket1785 170 31385-31402
START_RANGE client16v socket1786 180 31402-31420
START_RANGE client16w socket1787 170 31420-31437
START_RANGE client16x socket1788 180 31437-31455
START_RANGE client16y socket1789 180 31455-31473
START_RANGE client16z socket1790 170 31473-31490
START_RANGE client16aa socket1791 180 31490-31508
START_RANGE client16ab socket1792 170 31508-31525

START_RANGE client17a socket1793 180 31525-31543
START_RANGE client17b socket1794 180 31543-31561
START_RANGE client17c socket1795 170 31561-31578
START_RANGE client17d socket1796 180 31578-31596
START_RANGE client17e socket1797 170 31596-31613
START_RANGE client17f socket1798 180 31613-31631
START_RANGE client17g socket1799 180 31631-31649
START_RANGE client17h socket1800 170 31649-31666
START_RANGE client17i socket1801 180 31666-31684
START_RANGE client17j socket1802 170 31684-31701
START_RANGE client17k socket1803 180 31701-31719
START_RANGE client17l socket1804 180 31719-31737
START_RANGE client17m socket1805 170 31737-31754
START_RANGE client17n socket1806 180 31754-31772
START_RANGE client17a socket1807 170 31772-31789
START_RANGE client17b socket1808 180 31789-31807
START_RANGE client17c socket1809 170 31807-31824
START_RANGE client17d socket1810 180 31824-31842
START_RANGE client17e socket1811 180 31842-31860
START_RANGE client17f socket1812 170 31860-31877
START_RANGE client17g socket1813 180 31877-31895
START_RANGE client17h socket1814 170 31895-31912
START_RANGE client17i socket1815 180 31912-31930
START_RANGE client17j socket1816 180 31930-31948
START_RANGE client17k socket1817 170 31948-31965
START_RANGE client17l socket1818 180 31965-31983
START_RANGE client17m socket1819 170 31983-32000
START_RANGE client17n socket1820 180 32000-32018
START_RANGE client17a socket1821 180 32018-32036
START_RANGE client17b socket1822 170 32036-32053
START_RANGE client17c socket1823 180 32053-32071
START_RANGE client17d socket1824 170 32071-32088
START_RANGE client17e socket1825 180 32088-32106
START_RANGE client17f socket1826 180 32106-32124
START_RANGE client17g socket1827 170 32124-32141
START_RANGE client17h socket1828 180 32141-32159
START_RANGE client17i socket1829 170 32159-32176
START_RANGE client17j socket1830 180 32176-32194
START_RANGE client17k socket1831 180 32194-32212
START_RANGE client17l socket1832 170 32212-32229
START_RANGE client17m socket1833 180 32229-32247
START_RANGE client17n socket1834 170 32247-32264
START_RANGE client17a socket1835 180 32264-32282
START_RANGE client17b socket1836 170 32282-32299
START_RANGE client17c socket1837 180 32299-32317
START_RANGE client17d socket1838 180 32317-32335
START_RANGE client17e socket1839 170 32335-32352
START_RANGE client17f socket1840 180 32352-32370
START_RANGE client17g socket1841 170 32370-32387
START_RANGE client17h socket1842 180 32387-32405
START_RANGE client17i socket1843 180 32405-32423
START_RANGE client17j socket1844 170 32423-32440
START_RANGE client17k socket1845 180 32440-32458
START_RANGE client17l socket1846 170 32458-32475
START_RANGE client17m socket1847 180 32475-32493
START_RANGE client17n socket1848 180 32493-32511

START_RANGE client17o socket1849 170 32511-32528
START_RANGE client17p socket1850 180 32528-32546
START_RANGE client17q socket1851 170 32546-32563
START_RANGE client17r socket1852 180 32563-32581
START_RANGE client17s socket1853 180 32581-32599
START_RANGE client17t socket1854 170 32599-32616
START_RANGE client17u socket1855 180 32616-32634
START_RANGE client17v socket1856 170 32634-32651
START_RANGE client17w socket1857 180 32651-32669
START_RANGE client17x socket1858 180 32669-32687
START_RANGE client17y socket1859 170 32687-32704
START_RANGE client17z socket1860 180 32704-32722
START_RANGE client17aa socket1861 170 32722-32739
START_RANGE client17ab socket1862 180 32739-32757
START_RANGE client17o socket1863 170 32757-32774
START_RANGE client17p socket1864 180 32774-32792
START_RANGE client17q socket1865 180 32792-32810
START_RANGE client17r socket1866 170 32810-32827
START_RANGE client17s socket1867 180 32827-32845
START_RANGE client17t socket1868 170 32845-32862
START_RANGE client17u socket1869 180 32862-32880
START_RANGE client17v socket1870 180 32880-32898
START_RANGE client17w socket1871 170 32898-32915
START_RANGE client17x socket1872 180 32915-32933
START_RANGE client17y socket1873 170 32933-32950
START_RANGE client17z socket1874 180 32950-32968
START_RANGE client17aa socket1875 180 32968-32986
START_RANGE client17ab socket1876 170 32986-33003
START_RANGE client17o socket1877 180 33003-33021
START_RANGE client17p socket1878 170 33021-33038
START_RANGE client17q socket1879 180 33038-33056
START_RANGE client17r socket1880 180 33056-33074
START_RANGE client17s socket1881 170 33074-33091
START_RANGE client17t socket1882 180 33091-33109
START_RANGE client17u socket1883 170 33109-33126
START_RANGE client17v socket1884 180 33126-33144
START_RANGE client17w socket1885 180 33144-33162
START_RANGE client17x socket1886 170 33162-33179
START_RANGE client17y socket1887 180 33179-33197
START_RANGE client17z socket1888 170 33197-33214
START_RANGE client17aa socket1889 180 33214-33232
START_RANGE client17ab socket1890 170 33232-33249
START_RANGE client17o socket1891 180 33249-33267
START_RANGE client17p socket1892 180 33267-33285

START_RANGE client17q socket1893 170 33285-33302
START_RANGE client17r socket1894 180 33302-33320
START_RANGE client17s socket1895 170 33320-33337
START_RANGE client17t socket1896 180 33337-33355
START_RANGE client17u socket1897 180 33355-33373
START_RANGE client17v socket1898 170 33373-33390
START_RANGE client17w socket1899 180 33390-33408
START_RANGE client17x socket1900 170 33408-33425
START_RANGE client17y socket1901 180 33425-33443
START_RANGE client17z socket1902 180 33443-33461
START_RANGE client17aa socket1903 170 33461-33478
START_RANGE client17ab socket1904 180 33478-33496

START_RANGE client18a socket1905 170 33496-33513
START_RANGE client18b socket1906 180 33513-33531
START_RANGE client18c socket1907 180 33531-33549
START_RANGE client18d socket1908 170 33549-33566
START_RANGE client18e socket1909 180 33566-33584
START_RANGE client18f socket1910 170 33584-33601
START_RANGE client18g socket1911 180 33601-33619
START_RANGE client18h socket1912 180 33619-33637
START_RANGE client18i socket1913 170 33637-33654
START_RANGE client18j socket1914 180 33654-33672
START_RANGE client18k socket1915 170 33672-33689
START_RANGE client18l socket1916 180 33689-33707
START_RANGE client18m socket1917 170 33707-33724
START_RANGE client18n socket1918 180 33724-33742
START_RANGE client18a socket1919 180 33742-33760
START_RANGE client18b socket1920 170 33760-33777
START_RANGE client18c socket1921 180 33777-33795
START_RANGE client18d socket1922 170 33795-33812
START_RANGE client18e socket1923 180 33812-33830
START_RANGE client18f socket1924 180 33830-33848
START_RANGE client18g socket1925 170 33848-33865
START_RANGE client18h socket1926 180 33865-33883
START_RANGE client18i socket1927 170 33883-33900
START_RANGE client18j socket1928 180 33900-33918
START_RANGE client18k socket1929 180 33918-33936
START_RANGE client18l socket1930 170 33936-33953
START_RANGE client18m socket1931 180 33953-33971
START_RANGE client18n socket1932 170 33971-33988
START_RANGE client18a socket1933 180 33988-34006
START_RANGE client18b socket1934 180 34006-34024
START_RANGE client18c socket1935 170 34024-34041
START_RANGE client18d socket1936 180 34041-34059
START_RANGE client18e socket1937 170 34059-34076
START_RANGE client18f socket1938 180 34076-34094
START_RANGE client18g socket1939 180 34094-34112
START_RANGE client18h socket1940 170 34112-34129
START_RANGE client18i socket1941 180 34129-34147
START_RANGE client18j socket1942 170 34147-34164
START_RANGE client18k socket1943 180 34164-34182
START_RANGE client18l socket1944 170 34182-34199
START_RANGE client18m socket1945 180 34199-34217
START_RANGE client18n socket1946 180 34217-34235
START_RANGE client18a socket1947 170 34235-34252
START_RANGE client18b socket1948 180 34252-34270
START_RANGE client18c socket1949 170 34270-34287
START_RANGE client18d socket1950 180 34287-34305
START_RANGE client18e socket1951 180 34305-34323
START_RANGE client18f socket1952 170 34323-34340
START_RANGE client18g socket1953 180 34340-34358
START_RANGE client18h socket1954 170 34358-34375
START_RANGE client18i socket1955 180 34375-34393
START_RANGE client18j socket1956 180 34393-34411
START_RANGE client18k socket1957 170 34411-34428
START_RANGE client18l socket1958 180 34428-34446
START_RANGE client18m socket1959 170 34446-34463
START_RANGE client18n socket1960 180 34463-34481

START_RANGE client18o socket1961 180 34481-34499
START_RANGE client18p socket1962 170 34499-34516
START_RANGE client18q socket1963 180 34516-34534
START_RANGE client18r socket1964 170 34534-34551
START_RANGE client18s socket1965 180 34551-34569
START_RANGE client18t socket1966 180 34569-34587
START_RANGE client18u socket1967 170 34587-34604
START_RANGE client18v socket1968 180 34604-34622
START_RANGE client18w socket1969 170 34622-34639
START_RANGE client18x socket1970 180 34639-34657
START_RANGE client18y socket1971 170 34657-34674
START_RANGE client18z socket1972 180 34674-34692
START_RANGE client18aa socket1973 180 34692-34710
START_RANGE client18ab socket1974 170 34710-34727
START_RANGE client18o socket1975 180 34727-34745
START_RANGE client18p socket1976 170 34745-34762
START_RANGE client18q socket1977 180 34762-34780
START_RANGE client18r socket1978 180 34780-34798
START_RANGE client18s socket1979 170 34798-34815
START_RANGE client18t socket1980 180 34815-34833
START_RANGE client18u socket1981 170 34833-34850
START_RANGE client18v socket1982 180 34850-34868
START_RANGE client18w socket1983 180 34868-34886
START_RANGE client18x socket1984 170 34886-34903
START_RANGE client18y socket1985 180 34903-34921
START_RANGE client18z socket1986 170 34921-34938
START_RANGE client18aa socket1987 180 34938-34956
START_RANGE client18ab socket1988 180 34956-34974
START_RANGE client18o socket1989 170 34974-34991
START_RANGE client18p socket1990 180 34991-35009
START_RANGE client18q socket1991 170 35009-35026
START_RANGE client18r socket1992 180 35026-35044
START_RANGE client18s socket1993 180 35044-35062
START_RANGE client18t socket1994 170 35062-35079
START_RANGE client18u socket1995 180 35079-35097
START_RANGE client18v socket1996 170 35097-35114
START_RANGE client18w socket1997 180 35114-35132
START_RANGE client18x socket1998 170 35132-35149
START_RANGE client18y socket1999 180 35149-35167
START_RANGE client18z socket2000 180 35167-35185
START_RANGE client18aa socket2001 170 35185-35202
START_RANGE client18ab socket2002 180 35202-35220
START_RANGE client18o socket2003 170 35220-35237
START_RANGE client18p socket2004 180 35237-35255
START_RANGE client18q socket2005 180 35255-35273
START_RANGE client18r socket2006 170 35273-35290
START_RANGE client18s socket2007 180 35290-35308
START_RANGE client18t socket2008 170 35308-35325
START_RANGE client18u socket2009 180 35325-35343
START_RANGE client18v socket2010 180 35343-35361
START_RANGE client18w socket2011 170 35361-35378
START_RANGE client18x socket2012 180 35378-35396
START_RANGE client18y socket2013 170 35396-35413
START_RANGE client18z socket2014 180 35413-35431
START_RANGE client18aa socket2015 180 35431-35449
START_RANGE client18ab socket2016 170 35449-35466

#elif MASTER_NUM7
START_RANGE client19a socket2017 180 35466-35484
START_RANGE client19b socket2018 170 35484-35501
START_RANGE client19c socket2019 180 35501-35519
START_RANGE client19d socket2020 180 35519-35537
START_RANGE client19e socket2021 170 35537-35554
START_RANGE client19f socket2022 180 35554-35572
START_RANGE client19g socket2023 170 35572-35589
START_RANGE client19h socket2024 180 35589-35607
START_RANGE client19i socket2025 170 35607-35624
START_RANGE client19j socket2026 180 35624-35642
START_RANGE client19k socket2027 180 35642-35660
START_RANGE client19l socket2028 170 35660-35677
START_RANGE client19m socket2029 180 35677-35695
START_RANGE client19n socket2030 170 35695-35712
START_RANGE client19a socket2031 180 35712-35730
START_RANGE client19b socket2032 180 35730-35748
START_RANGE client19c socket2033 170 35748-35765
START_RANGE client19d socket2034 180 35765-35783
START_RANGE client19e socket2035 170 35783-35800
START_RANGE client19f socket2036 180 35800-35818
START_RANGE client19g socket2037 180 35818-35836
START_RANGE client19h socket2038 170 35836-35853
START_RANGE client19i socket2039 180 35853-35871
START_RANGE client19j socket2040 170 35871-35888
START_RANGE client19k socket2041 180 35888-35906
START_RANGE client19l socket2042 180 35906-35924
START_RANGE client19m socket2043 170 35924-35941
START_RANGE client19n socket2044 180 35941-35959
START_RANGE client19a socket2045 170 35959-35976
START_RANGE client19b socket2046 180 35976-35994
START_RANGE client19c socket2047 180 35994-36012
START_RANGE client19d socket2048 170 36012-36029
START_RANGE client19e socket2049 180 36029-36047
START_RANGE client19f socket2050 170 36047-36064
START_RANGE client19g socket2051 180 36064-36082
START_RANGE client19h socket2052 170 36082-36099
START_RANGE client19i socket2053 180 36099-36117
START_RANGE client19j socket2054 180 36117-36135
START_RANGE client19k socket2055 170 36135-36152
START_RANGE client19l socket2056 180 36152-36170
START_RANGE client19m socket2057 170 36170-36187
START_RANGE client19n socket2058 180 36187-36205
START_RANGE client19a socket2059 180 36205-36223
START_RANGE client19b socket2060 170 36223-36240
START_RANGE client19c socket2061 180 36240-36258
START_RANGE client19d socket2062 170 36258-36275
START_RANGE client19e socket2063 180 36275-36293
START_RANGE client19f socket2064 180 36293-36311
START_RANGE client19g socket2065 170 36311-36328
START_RANGE client19h socket2066 180 36328-36346
START_RANGE client19i socket2067 170 36346-36363
START_RANGE client19j socket2068 180 36363-36381
START_RANGE client19k socket2069 180 36381-36399
START_RANGE client19l socket2070 170 36399-36416
START_RANGE client19m socket2071 180 36416-36434
START_RANGE client19n socket2072 170 36434-36451

START_RANGE client19o socket2073 180 36451-36469
START_RANGE client19p socket2074 180 36469-36487
START_RANGE client19q socket2075 170 36487-36504
START_RANGE client19r socket2076 180 36504-36522
START_RANGE client19s socket2077 170 36522-36539
START_RANGE client19t socket2078 180 36539-36557
START_RANGE client19u socket2079 170 36557-36574
START_RANGE client19v socket2080 180 36574-36592
START_RANGE client19w socket2081 180 36592-36610
START_RANGE client19x socket2082 170 36610-36627
START_RANGE client19y socket2083 180 36627-36645
START_RANGE client19z socket2084 170 36645-36662
START_RANGE client19aa socket2085 180 36662-36680

START_RANGE client19ab socket2086 180 36680-36698
START_RANGE client19o socket2087 170 36698-36715
START_RANGE client19p socket2088 180 36715-36733
START_RANGE client19q socket2089 170 36733-36750
START_RANGE client19r socket2090 180 36750-36768
START_RANGE client19s socket2091 180 36768-36786
START_RANGE client19t socket2092 170 36786-36803
START_RANGE client19u socket2093 180 36803-36821
START_RANGE client19v socket2094 170 36821-36838
START_RANGE client19w socket2095 180 36838-36856
START_RANGE client19x socket2096 180 36856-36874
START_RANGE client19y socket2097 170 36874-36891
START_RANGE client19z socket2098 180 36891-36909
START_RANGE client19aa socket2099 170 36909-36926
START_RANGE client19ab socket2100 180 36926-36944
START_RANGE client19o socket2101 180 36944-36962
START_RANGE client19p socket2102 170 36962-36979
START_RANGE client19q socket2103 180 36979-36997
START_RANGE client19r socket2104 170 36997-37014
START_RANGE client19s socket2105 180 37014-37032
START_RANGE client19t socket2106 170 37032-37049
START_RANGE client19u socket2107 180 37049-37067
START_RANGE client19v socket2108 180 37067-37085
START_RANGE client19w socket2109 170 37085-37102
START_RANGE client19x socket2110 180 37102-37120
START_RANGE client19y socket2111 170 37120-37137
START_RANGE client19z socket2112 180 37137-37155
START_RANGE client19aa socket2113 180 37155-37173
START_RANGE client19ab socket2114 170 37173-37190
START_RANGE client19o socket2115 180 37190-37208
START_RANGE client19p socket2116 170 37208-37225
START_RANGE client19q socket2117 180 37225-37243
START_RANGE client19r socket2118 180 37243-37261
START_RANGE client19s socket2119 170 37261-37278
START_RANGE client19t socket2120 180 37278-37296
START_RANGE client19u socket2121 170 37296-37313
START_RANGE client19v socket2122 180 37313-37331
START_RANGE client19w socket2123 180 37331-37349
START_RANGE client19x socket2124 170 37349-37366
START_RANGE client19y socket2125 180 37366-37384
START_RANGE client19z socket2126 170 37384-37401
START_RANGE client19aa socket2127 180 37401-37419
START_RANGE client19ab socket2128 180 37419-37437

START_RANGE client20a socket2129 170 37437-37454
START_RANGE client20b socket2130 180 37454-37472
START_RANGE client20c socket2131 170 37472-37489
START_RANGE client20d socket2132 180 37489-37507
START_RANGE client20e socket2133 170 37507-37524
START_RANGE client20f socket2134 180 37524-37542
START_RANGE client20g socket2135 180 37542-37560
START_RANGE client20h socket2136 170 37560-37577
START_RANGE client20i socket2137 180 37577-37595
START_RANGE client20j socket2138 170 37595-37612
START_RANGE client20k socket2139 180 37612-37630
START_RANGE client20l socket2140 180 37630-37648
START_RANGE client20m socket2141 170 37648-37665
START_RANGE client20n socket2142 180 37665-37683
START_RANGE client20a socket2143 170 37683-37700
START_RANGE client20b socket2144 180 37700-37718
START_RANGE client20c socket2145 180 37718-37736
START_RANGE client20d socket2146 170 37736-37753
START_RANGE client20e socket2147 180 37753-37771
START_RANGE client20f socket2148 170 37771-37788
START_RANGE client20g socket2149 180 37788-37806
START_RANGE client20h socket2150 180 37806-37824
START_RANGE client20i socket2151 170 37824-37841
START_RANGE client20j socket2152 180 37841-37859
START_RANGE client20k socket2153 170 37859-37876
START_RANGE client20l socket2154 180 37876-37894
START_RANGE client20m socket2155 180 37894-37912
START_RANGE client20n socket2156 170 37912-37929
START_RANGE client20a socket2157 180 37929-37947
START_RANGE client20b socket2158 170 37947-37964
START_RANGE client20c socket2159 180 37964-37982
START_RANGE client20d socket2160 170 37982-37999
START_RANGE client20e socket2161 180 37999-38017
START_RANGE client20f socket2162 180 38017-38035
START_RANGE client20g socket2163 170 38035-38052
START_RANGE client20h socket2164 180 38052-38070
START_RANGE client20i socket2165 170 38070-38087
START_RANGE client20j socket2166 180 38087-38105
START_RANGE client20k socket2167 180 38105-38123
START_RANGE client20l socket2168 170 38123-38140
START_RANGE client20m socket2169 180 38140-38158
START_RANGE client20n socket2170 170 38158-38175
START_RANGE client20a socket2171 180 38175-38193
START_RANGE client20b socket2172 180 38193-38211
START_RANGE client20c socket2173 170 38211-38228
START_RANGE client20d socket2174 180 38228-38246
START_RANGE client20e socket2175 170 38246-38263
START_RANGE client20f socket2176 180 38263-38281
START_RANGE client20g socket2177 180 38281-38299
START_RANGE client20h socket2178 170 38299-38316
START_RANGE client20i socket2179 180 38316-38334
START_RANGE client20j socket2180 170 38334-38351
START_RANGE client20k socket2181 180 38351-38369
START_RANGE client20l socket2182 180 38369-38387
START_RANGE client20m socket2183 170 38387-38404
START_RANGE client20n socket2184 180 38404-38422

START_RANGE client20o socket2185 170 38422-38439
START_RANGE client20p socket2186 180 38439-38457
START_RANGE client20q socket2187 170 38457-38474
START_RANGE client20r socket2188 180 38474-38492
START_RANGE client20s socket2189 180 38492-38510
START_RANGE client20t socket2190 170 38510-38527
START_RANGE client20u socket2191 180 38527-38545
START_RANGE client20v socket2192 170 38545-38562
START_RANGE client20w socket2193 180 38562-38580
START_RANGE client20x socket2194 180 38580-38598
START_RANGE client20y socket2195 170 38598-38615
START_RANGE client20z socket2196 180 38615-38633
START_RANGE client20aa socket2197 170 38633-38650
START_RANGE client20ab socket2198 180 38650-38668
START_RANGE client20o socket2199 180 38668-38686
START_RANGE client20p socket2200 170 38686-38703
START_RANGE client20q socket2201 180 38703-38721
START_RANGE client20r socket2202 170 38721-38738
START_RANGE client20s socket2203 180 38738-38756
START_RANGE client20t socket2204 180 38756-38774
START_RANGE client20u socket2205 170 38774-38791
START_RANGE client20v socket2206 180 38791-38809
START_RANGE client20w socket2207 170 38809-38826
START_RANGE client20x socket2208 180 38826-38844
START_RANGE client20y socket2209 180 38844-38862
START_RANGE client20z socket2210 170 38862-38879
START_RANGE client20aa socket2211 180 38879-38897
START_RANGE client20ab socket2212 170 38897-38914
START_RANGE client20o socket2213 180 38914-38932
START_RANGE client20p socket2214 170 38932-38949
START_RANGE client20q socket2215 180 38949-38967
START_RANGE client20r socket2216 180 38967-38985
START_RANGE client20s socket2217 170 38985-39002
START_RANGE client20t socket2218 180 39002-39020
START_RANGE client20u socket2219 170 39020-39037
START_RANGE client20v socket2220 180 39037-39055
START_RANGE client20w socket2221 180 39055-39073
START_RANGE client20x socket2222 170 39073-39090
START_RANGE client20y socket2223 180 39090-39108
START_RANGE client20z socket2224 170 39108-39125
START_RANGE client20aa socket2225 180 39125-39143
START_RANGE client20ab socket2226 180 39143-39161
START_RANGE client20o socket2227 170 39161-39178
START_RANGE client20p socket2228 180 39178-39196
START_RANGE client20q socket2229 170 39196-39213
START_RANGE client20r socket2230 180 39213-39231
START_RANGE client20s socket2231 180 39231-39249
START_RANGE client20t socket2232 170 39249-39266
START_RANGE client20u socket2233 180 39266-39284
START_RANGE client20v socket2234 170 39284-39301
START_RANGE client20w socket2235 180 39301-39319
START_RANGE client20x socket2236 180 39319-39337
START_RANGE client20y socket2237 170 39337-39354
START_RANGE client20z socket2238 180 39354-39372
START_RANGE client20aa socket2239 170 39372-39389
START_RANGE client20ab socket2240 180 39389-39407

START_RANGE client21a socket2241 170 39407-39424
START_RANGE client21b socket2242 180 39424-39442
START_RANGE client21c socket2243 180 39442-39460
START_RANGE client21d socket2244 170 39460-39477
START_RANGE client21e socket2245 180 39477-39495
START_RANGE client21f socket2246 170 39495-39512
START_RANGE client21g socket2247 180 39512-39530
START_RANGE client21h socket2248 180 39530-39548
START_RANGE client21i socket2249 170 39548-39565
START_RANGE client21j socket2250 180 39565-39583
START_RANGE client21k socket2251 170 39583-39600
START_RANGE client21l socket2252 180 39600-39618
START_RANGE client21m socket2253 180 39618-39636
START_RANGE client21n socket2254 170 39636-39653
START_RANGE client21a socket2255 180 39653-39671
START_RANGE client21b socket2256 170 39671-39688
START_RANGE client21c socket2257 180 39688-39706
START_RANGE client21d socket2258 180 39706-39724
START_RANGE client21e socket2259 170 39724-39741
START_RANGE client21f socket2260 180 39741-39759
START_RANGE client21g socket2261 170 39759-39776
START_RANGE client21h socket2262 180 39776-39794
START_RANGE client21i socket2263 180 39794-39812
START_RANGE client21j socket2264 170 39812-39829
START_RANGE client21k socket2265 180 39829-39847
START_RANGE client21l socket2266 170 39847-39864
START_RANGE client21m socket2267 180 39864-39882
START_RANGE client21n socket2268 170 39882-39899
START_RANGE client21a socket2269 180 39899-39917
START_RANGE client21b socket2270 180 39917-39935
START_RANGE client21c socket2271 170 39935-39952
START_RANGE client21d socket2272 180 39952-39970
START_RANGE client21e socket2273 170 39970-39987
START_RANGE client21f socket2274 180 39987-40005
START_RANGE client21g socket2275 180 40005-40023
START_RANGE client21h socket2276 170 40023-40040
START_RANGE client21i socket2277 180 40040-40058
START_RANGE client21j socket2278 170 40058-40075
START_RANGE client21k socket2279 180 40075-40093
START_RANGE client21l socket2280 180 40093-40111

```
START_RANGE client21m socket2281 170 40111-40128
START_RANGE client21n socket2282 180 40128-40146
START_RANGE client21a socket2283 170 40146-40163
START_RANGE client21b socket2284 180 40163-40181
START_RANGE client21c socket2285 180 40181-40199
START_RANGE client21d socket2286 170 40199-40216
START_RANGE client21e socket2287 180 40216-40234
START_RANGE client21f socket2288 170 40234-40251
START_RANGE client21g socket2289 180 40251-40269
START_RANGE client21h socket2290 180 40269-40287
START_RANGE client21i socket2291 170 40287-40304
START_RANGE client21j socket2292 180 40304-40322
START_RANGE client21k socket2293 170 40322-40339
START_RANGE client21l socket2294 180 40339-40357
START_RANGE client21m socket2295 170 40357-40374
START_RANGE client21n socket2296 180 40374-40392

START_RANGE client21o socket2297 180 40392-40410
START_RANGE client21p socket2298 170 40410-40427
START_RANGE client21q socket2299 180 40427-40445
START_RANGE client21r socket2300 170 40445-40462
START_RANGE client21s socket2301 180 40462-40480
START_RANGE client21t socket2302 180 40480-40498
START_RANGE client21u socket2303 170 40498-40515
START_RANGE client21v socket2304 180 40515-40533
START_RANGE client21w socket2305 170 40533-40550
START_RANGE client21x socket2306 180 40550-40568
START_RANGE client21y socket2307 180 40568-40586
START_RANGE client21z socket2308 170 40586-40603
START_RANGE client21aa socket2309 180 40603-40621
START_RANGE client21ab socket2310 170 40621-40638
START_RANGE client21o socket2311 180 40638-40656
START_RANGE client21p socket2312 180 40656-40674
START_RANGE client21q socket2313 170 40674-40691
START_RANGE client21r socket2314 180 40691-40709
START_RANGE client21s socket2315 170 40709-40726
START_RANGE client21t socket2316 180 40726-40744
START_RANGE client21u socket2317 180 40744-40762
START_RANGE client21v socket2318 170 40762-40779
START_RANGE client21w socket2319 180 40779-40797
START_RANGE client21x socket2320 170 40797-40814
START_RANGE client21y socket2321 180 40814-40832
START_RANGE client21z socket2322 170 40832-40849
START_RANGE client21aa socket2323 180 40849-40867
START_RANGE client21ab socket2324 180 40867-40885
START_RANGE client21o socket2325 170 40885-40902
START_RANGE client21p socket2326 180 40902-40920
START_RANGE client21q socket2327 170 40920-40937
START_RANGE client21r socket2328 180 40937-40955
START_RANGE client21s socket2329 180 40955-40973
START_RANGE client21t socket2330 170 40973-40990
START_RANGE client21u socket2331 180 40990-41008
START_RANGE client21v socket2332 170 41008-41025
START_RANGE client21w socket2333 180 41025-41043
START_RANGE client21x socket2334 180 41043-41061
START_RANGE client21y socket2335 170 41061-41078
START_RANGE client21z socket2336 180 41078-41096
START_RANGE client21aa socket2337 170 41096-41113
START_RANGE client21ab socket2338 180 41113-41131
START_RANGE client21o socket2339 180 41131-41149
START_RANGE client21p socket2340 170 41149-41166
START_RANGE client21q socket2341 180 41166-41184
START_RANGE client21r socket2342 170 41184-41201
START_RANGE client21s socket2343 180 41201-41219
START_RANGE client21t socket2344 180 41219-41237
START_RANGE client21u socket2345 170 41237-41254
START_RANGE client21v socket2346 180 41254-41272
START_RANGE client21w socket2347 170 41272-41289
START_RANGE client21x socket2348 180 41289-41307
START_RANGE client21y socket2349 170 41307-41324
START_RANGE client21z socket2350 180 41324-41342
START_RANGE client21aa socket2351 180 41342-41360
START_RANGE client21ab socket2352 170 41360-41377

#elif MASTER_NUM8
START_RANGE client22a socket2353 180 41377-41395
START_RANGE client22b socket2354 170 41395-41412
START_RANGE client22c socket2355 180 41412-41430
START_RANGE client22d socket2356 180 41430-41448
START_RANGE client22e socket2357 170 41448-41465
START_RANGE client22f socket2358 180 41465-41483
START_RANGE client22g socket2359 170 41483-41500
START_RANGE client22h socket2360 180 41500-41518
START_RANGE client22i socket2361 180 41518-41536
START_RANGE client22j socket2362 170 41536-41553
START_RANGE client22k socket2363 180 41553-41571
START_RANGE client22l socket2364 170 41571-41588
START_RANGE client22m socket2365 180 41588-41606
START_RANGE client22n socket2366 180 41606-41624
START_RANGE client22a socket2367 170 41624-41641
START_RANGE client22b socket2368 180 41641-41659
START_RANGE client22c socket2369 170 41659-41676
START_RANGE client22d socket2370 180 41676-41694
START_RANGE client22e socket2371 180 41694-41712
START_RANGE client22f socket2372 170 41712-41729
START_RANGE client22g socket2373 180 41729-41747
START_RANGE client22h socket2374 170 41747-41764
START_RANGE client22i socket2375 180 41764-41782
START_RANGE client22j socket2376 170 41782-41799
START_RANGE client22k socket2377 180 41799-41817
START_RANGE client22l socket2378 180 41817-41835
START_RANGE client22m socket2379 170 41835-41852
START_RANGE client22n socket2380 180 41852-41870
START_RANGE client22a socket2381 170 41870-41887
START_RANGE client22b socket2382 180 41887-41905
START_RANGE client22c socket2383 180 41905-41923
START_RANGE client22d socket2384 170 41923-41940
START_RANGE client22e socket2385 180 41940-41958
START_RANGE client22f socket2386 170 41958-41975
START_RANGE client22g socket2387 180 41975-41993
START_RANGE client22h socket2388 180 41993-42011
START_RANGE client22i socket2389 170 42011-42028
START_RANGE client22j socket2390 180 42028-42046
START_RANGE client22k socket2391 170 42046-42063
START_RANGE client22l socket2392 180 42063-42081
START_RANGE client22m socket2393 180 42081-42099
START_RANGE client22n socket2394 170 42099-42116
START_RANGE client22a socket2395 180 42116-42134
START_RANGE client22b socket2396 170 42134-42151
START_RANGE client22c socket2397 180 42151-42169
START_RANGE client22d socket2398 180 42169-42187
START_RANGE client22e socket2399 170 42187-42204
START_RANGE client22f socket2400 180 42204-42222
START_RANGE client22g socket2401 170 42222-42239
START_RANGE client22h socket2402 180 42239-42257
START_RANGE client22i socket2403 170 42257-42274
START_RANGE client22j socket2404 180 42274-42292
START_RANGE client22k socket2405 180 42292-42310
START_RANGE client22l socket2406 170 42310-42327
START_RANGE client22m socket2407 180 42327-42345
START_RANGE client22n socket2408 170 42345-42362

START_RANGE client22o socket2409 180 42362-42380
START_RANGE client22p socket2410 180 42380-42398
START_RANGE client22q socket2411 170 42398-42415
START_RANGE client22r socket2412 180 42415-42433
START_RANGE client22s socket2413 170 42433-42450
START_RANGE client22t socket2414 180 42450-42468
START_RANGE client22u socket2415 180 42468-42486
START_RANGE client22v socket2416 170 42486-42503
START_RANGE client22w socket2417 180 42503-42521
START_RANGE client22x socket2418 170 42521-42538
START_RANGE client22y socket2419 180 42538-42556
START_RANGE client22z socket2420 180 42556-42574
START_RANGE client22aa socket2421 170 42574-42591
START_RANGE client22ab socket2422 180 42591-42609
START_RANGE client22o socket2423 170 42609-42626
START_RANGE client22p socket2424 180 42626-42644
START_RANGE client22q socket2425 180 42644-42662
START_RANGE client22r socket2426 170 42662-42679
START_RANGE client22s socket2427 180 42679-42697
START_RANGE client22t socket2428 170 42697-42714
START_RANGE client22u socket2429 180 42714-42732
START_RANGE client22v socket2430 170 42732-42749
START_RANGE client22w socket2431 180 42749-42767
START_RANGE client22x socket2432 180 42767-42785
START_RANGE client22y socket2433 170 42785-42802
START_RANGE client22z socket2434 180 42802-42820
START_RANGE client22aa socket2435 170 42820-42837
START_RANGE client22ab socket2436 180 42837-42855
START_RANGE client22o socket2437 180 42855-42873
START_RANGE client22p socket2438 170 42873-42890
START_RANGE client22q socket2439 180 42890-42908
START_RANGE client22r socket2440 170 42908-42925
START_RANGE client22s socket2441 180 42925-42943
START_RANGE client22t socket2442 180 42943-42961
START_RANGE client22u socket2443 170 42961-42978
START_RANGE client22v socket2444 180 42978-42996
START_RANGE client22w socket2445 170 42996-43013
START_RANGE client22x socket2446 180 43013-43031
START_RANGE client22y socket2447 180 43031-43049
START_RANGE client22z socket2448 170 43049-43066
START_RANGE client22aa socket2449 180 43066-43084
START_RANGE client22ab socket2450 170 43084-43101
START_RANGE client22o socket2451 180 43101-43119
START_RANGE client22p socket2452 180 43119-43137
START_RANGE client22q socket2453 170 43137-43154
START_RANGE client22r socket2454 180 43154-43172
START_RANGE client22s socket2455 170 43172-43189
START_RANGE client22t socket2456 180 43189-43207
START_RANGE client22u socket2457 170 43207-43224
START_RANGE client22v socket2458 180 43224-43242
START_RANGE client22w socket2459 180 43242-43260
START_RANGE client22x socket2460 170 43260-43277
START_RANGE client22y socket2461 180 43277-43295
START_RANGE client22z socket2462 170 43295-43312
START_RANGE client22aa socket2463 180 43312-43330
START_RANGE client22ab socket2464 180 43330-43348

START_RANGE client23a socket2465 170 43348-43365
START_RANGE client23b socket2466 180 43365-43383
START_RANGE client23c socket2467 170 43383-43400
START_RANGE client23d socket2468 180 43400-43418
START_RANGE client23e socket2469 180 43418-43436
START_RANGE client23f socket2470 170 43436-43453
START_RANGE client23g socket2471 180 43453-43471
START_RANGE client23h socket2472 170 43471-43488
START_RANGE client23i socket2473 180 43488-43506
```

START_RANGE client23j socket2474 180 43506-43524
START_RANGE client23k socket2475 170 43524-43541
START_RANGE client23l socket2476 180 43541-43559
START_RANGE client23m socket2477 170 43559-43576
START_RANGE client23n socket2478 180 43576-43594
START_RANGE client23a socket2479 180 43594-43612
START_RANGE client23b socket2480 170 43612-43629
START_RANGE client23c socket2481 180 43629-43647
START_RANGE client23d socket2482 170 43647-43664
START_RANGE client23e socket2483 180 43664-43682
START_RANGE client23f socket2484 170 43682-43699
START_RANGE client23g socket2485 180 43699-43717
START_RANGE client23h socket2486 180 43717-43735
START_RANGE client23i socket2487 170 43735-43752
START_RANGE client23j socket2488 180 43752-43770
START_RANGE client23k socket2489 170 43770-43787
START_RANGE client23l socket2490 180 43787-43805
START_RANGE client23m socket2491 180 43805-43823
START_RANGE client23n socket2492 170 43823-43840
START_RANGE client23a socket2493 180 43840-43858
START_RANGE client23b socket2494 170 43858-43875
START_RANGE client23c socket2495 180 43875-43893
START_RANGE client23d socket2496 180 43893-43911
START_RANGE client23e socket2497 170 43911-43928
START_RANGE client23f socket2498 180 43928-43946
START_RANGE client23g socket2499 170 43946-43963
START_RANGE client23h socket2500 180 43963-43981
START_RANGE client23i socket2501 180 43981-43999
START_RANGE client23j socket2502 170 43999-44016
START_RANGE client23k socket2503 180 44016-44034
START_RANGE client23l socket2504 170 44034-44051
START_RANGE client23m socket2505 180 44051-44069
START_RANGE client23n socket2506 180 44069-44087
START_RANGE client23a socket2507 170 44087-44104
START_RANGE client23b socket2508 180 44104-44122
START_RANGE client23c socket2509 170 44122-44139
START_RANGE client23d socket2510 180 44139-44157
START_RANGE client23e socket2511 170 44157-44174
START_RANGE client23f socket2512 180 44174-44192
START_RANGE client23g socket2513 180 44192-44210
START_RANGE client23h socket2514 170 44210-44227
START_RANGE client23i socket2515 180 44227-44245
START_RANGE client23j socket2516 170 44245-44262
START_RANGE client23k socket2517 180 44262-44280
START_RANGE client23l socket2518 180 44280-44298
START_RANGE client23m socket2519 170 44298-44315
START_RANGE client23n socket2520 180 44315-44333

START_RANGE client23o socket2521 170 44333-44350
START_RANGE client23p socket2522 180 44350-44368
START_RANGE client23q socket2523 180 44368-44386
START_RANGE client23r socket2524 170 44386-44403
START_RANGE client23s socket2525 180 44403-44421
START_RANGE client23t socket2526 170 44421-44438
START_RANGE client23u socket2527 180 44438-44456
START_RANGE client23v socket2528 180 44456-44474
START_RANGE client23w socket2529 170 44474-44491
START_RANGE client23x socket2530 180 44491-44509
START_RANGE client23y socket2531 170 44509-44526
START_RANGE client23z socket2532 180 44526-44544
START_RANGE client23aa socket2533 180 44544-44562
START_RANGE client23ab socket2534 170 44562-44579
START_RANGE client23o socket2535 180 44579-44597
START_RANGE client23p socket2536 170 44597-44614
START_RANGE client23q socket2537 180 44614-44632
START_RANGE client23r socket2538 170 44632-44649
START_RANGE client23s socket2539 180 44649-44667
START_RANGE client23t socket2540 180 44667-44685
START_RANGE client23u socket2541 170 44685-44702
START_RANGE client23v socket2542 180 44702-44720
START_RANGE client23w socket2543 170 44720-44737
START_RANGE client23x socket2544 180 44737-44755
START_RANGE client23y socket2545 180 44755-44773
START_RANGE client23z socket2546 170 44773-44790
START_RANGE client23aa socket2547 180 44790-44808
START_RANGE client23ab socket2548 170 44808-44825
START_RANGE client23o socket2549 180 44825-44843
START_RANGE client23p socket2550 180 44843-44861
START_RANGE client23q socket2551 170 44861-44878
START_RANGE client23r socket2552 180 44878-44896
START_RANGE client23s socket2553 170 44896-44913
START_RANGE client23t socket2554 180 44913-44931
START_RANGE client23u socket2555 180 44931-44949
START_RANGE client23v socket2556 170 44949-44966
START_RANGE client23w socket2557 180 44966-44984
START_RANGE client23x socket2558 170 44984-45001
START_RANGE client23y socket2559 180 45001-45019
START_RANGE client23z socket2560 180 45019-45037
START_RANGE client23aa socket2561 170 45037-45054
START_RANGE client23ab socket2562 180 45054-45072
START_RANGE client23o socket2563 170 45072-45089
START_RANGE client23p socket2564 180 45089-45107
START_RANGE client23q socket2565 170 45107-45124
START_RANGE client23r socket2566 180 45124-45142
START_RANGE client23s socket2567 180 45142-45160
START_RANGE client23t socket2568 170 45160-45177
START_RANGE client23u socket2569 180 45177-45195
START_RANGE client23v socket2570 170 45195-45212
START_RANGE client23w socket2571 180 45212-45230
START_RANGE client23x socket2572 180 45230-45248
START_RANGE client23y socket2573 170 45248-45265
START_RANGE client23z socket2574 180 45265-45283
START_RANGE client23aa socket2575 170 45283-45300
START_RANGE client23ab socket2576 180 45300-45318

START_RANGE client24a socket2577 180 45318-45336
START_RANGE client24b socket2578 170 45336-45353
START_RANGE client24c socket2579 180 45353-45371
START_RANGE client24d socket2580 170 45371-45388
START_RANGE client24e socket2581 180 45388-45406
START_RANGE client24f socket2582 180 45406-45424
START_RANGE client24g socket2583 170 45424-45441
START_RANGE client24h socket2584 180 45441-45459
START_RANGE client24i socket2585 170 45459-45476
START_RANGE client24j socket2586 180 45476-45494
START_RANGE client24k socket2587 180 45494-45512
START_RANGE client24l socket2588 170 45512-45529
START_RANGE client24m socket2589 180 45529-45547
START_RANGE client24n socket2590 170 45547-45564
START_RANGE client24a socket2591 180 45564-45582
START_RANGE client24b socket2592 170 45582-45599
START_RANGE client24c socket2593 180 45599-45617
START_RANGE client24d socket2594 180 45617-45635
START_RANGE client24e socket2595 170 45635-45652
START_RANGE client24f socket2596 180 45652-45670
START_RANGE client24g socket2597 170 45670-45687
START_RANGE client24h socket2598 180 45687-45705
START_RANGE client24i socket2599 180 45705-45723
START_RANGE client24j socket2600 170 45723-45740
START_RANGE client24k socket2601 180 45740-45758
START_RANGE client24l socket2602 170 45758-45775
START_RANGE client24m socket2603 180 45775-45793
START_RANGE client24n socket2604 180 45793-45811
START_RANGE client24a socket2605 170 45811-45828
START_RANGE client24b socket2606 180 45828-45846
START_RANGE client24c socket2607 170 45846-45863
START_RANGE client24d socket2608 180 45863-45881
START_RANGE client24e socket2609 180 45881-45899
START_RANGE client24f socket2610 170 45899-45916
START_RANGE client24g socket2611 180 45916-45934
START_RANGE client24h socket2612 170 45934-45951
START_RANGE client24i socket2613 180 45951-45969
START_RANGE client24j socket2614 180 45969-45987
START_RANGE client24k socket2615 170 45987-46004
START_RANGE client24l socket2616 180 46004-46022
START_RANGE client24m socket2617 170 46022-46039
START_RANGE client24n socket2618 180 46039-46057
START_RANGE client24a socket2619 170 46057-46074
START_RANGE client24b socket2620 180 46074-46092
START_RANGE client24c socket2621 180 46092-46110
START_RANGE client24d socket2622 170 46110-46127
START_RANGE client24e socket2623 180 46127-46145
START_RANGE client24f socket2624 170 46145-46162
START_RANGE client24g socket2625 180 46162-46180
START_RANGE client24h socket2626 180 46180-46198
START_RANGE client24i socket2627 170 46198-46215
START_RANGE client24j socket2628 180 46215-46233
START_RANGE client24k socket2629 170 46233-46250
START_RANGE client24l socket2630 180 46250-46268
START_RANGE client24m socket2631 180 46268-46286
START_RANGE client24n socket2632 170 46286-46303

START_RANGE client24o socket2633 180 46303-46321
START_RANGE client24p socket2634 170 46321-46338
START_RANGE client24q socket2635 180 46338-46356
START_RANGE client24r socket2636 180 46356-46374
START_RANGE client24s socket2637 170 46374-46391
START_RANGE client24t socket2638 180 46391-46409
START_RANGE client24u socket2639 170 46409-46426
START_RANGE client24v socket2640 180 46426-46444
START_RANGE client24w socket2641 180 46444-46462
START_RANGE client24x socket2642 170 46462-46479
START_RANGE client24y socket2643 180 46479-46497
START_RANGE client24z socket2644 170 46497-46514
START_RANGE client24aa socket2645 180 46514-46532
START_RANGE client24ab socket2646 170 46532-46549
START_RANGE client24o socket2647 180 46549-46567
START_RANGE client24p socket2648 180 46567-46585
START_RANGE client24q socket2649 170 46585-46602
START_RANGE client24r socket2650 180 46602-46620
START_RANGE client24s socket2651 170 46620-46637
START_RANGE client24t socket2652 180 46637-46655
START_RANGE client24u socket2653 180 46655-46673
START_RANGE client24v socket2654 170 46673-46690
START_RANGE client24w socket2655 180 46690-46708
START_RANGE client24x socket2656 170 46708-46725
START_RANGE client24y socket2657 180 46725-46743
START_RANGE client24z socket2658 180 46743-46761
START_RANGE client24aa socket2659 170 46761-46778
START_RANGE client24ab socket2660 180 46778-46796
START_RANGE client24o socket2661 170 46796-46813
START_RANGE client24p socket2662 180 46813-46831
START_RANGE client24q socket2663 180 46831-46849
START_RANGE client24r socket2664 170 46849-46866
START_RANGE client24s socket2665 180 46866-46884
START_RANGE client24t socket2666 170 46884-46901
START_RANGE client24u socket2667 180 46901-46919
START_RANGE client24v socket2668 180 46919-46937

START_RANGE client24w socket2669 170 46937-46954
START_RANGE client24x socket2670 180 46954-46972
START_RANGE client24y socket2671 170 46972-46989
START_RANGE client24z socket2672 180 46989-47007
START_RANGE client24aa socket2673 170 47007-47024
START_RANGE client24ab socket2674 180 47024-47042
START_RANGE client24o socket2675 180 47042-47060
START_RANGE client24p socket2676 170 47060-47077
START_RANGE client24q socket2677 180 47077-47095
START_RANGE client24r socket2678 170 47095-47112
START_RANGE client24s socket2679 180 47112-47130
START_RANGE client24t socket2680 180 47130-47148
START_RANGE client24u socket2681 170 47148-47165
START_RANGE client24v socket2682 180 47165-47183
START_RANGE client24w socket2683 170 47183-47200
START_RANGE client24x socket2684 180 47200-47218
START_RANGE client24y socket2685 180 47218-47236
START_RANGE client24z socket2686 170 47236-47253
START_RANGE client24aa socket2687 180 47253-47271
START_RANGE client24ab socket2688 170 47271-47288

#elif MASTER_NUM9
START_RANGE client25a socket2689 180 47288-47306
START_RANGE client25b socket2690 180 47306-47324
START_RANGE client25c socket2691 170 47324-47341
START_RANGE client25d socket2692 180 47341-47359
START_RANGE client25e socket2693 170 47359-47376
START_RANGE client25f socket2694 180 47376-47394
START_RANGE client25g socket2695 180 47394-47412
START_RANGE client25h socket2696 170 47412-47429
START_RANGE client25i socket2697 180 47429-47447
START_RANGE client25j socket2698 170 47447-47464
START_RANGE client25k socket2699 180 47464-47482
START_RANGE client25l socket2700 170 47482-47499
START_RANGE client25m socket2701 180 47499-47517
START_RANGE client25n socket2702 180 47517-47535
START_RANGE client25a socket2703 170 47535-47552
START_RANGE client25b socket2704 180 47552-47570
START_RANGE client25c socket2705 170 47570-47587
START_RANGE client25d socket2706 180 47587-47605
START_RANGE client25e socket2707 180 47605-47623
START_RANGE client25f socket2708 170 47623-47640
START_RANGE client25g socket2709 180 47640-47658
START_RANGE client25h socket2710 170 47658-47675
START_RANGE client25i socket2711 180 47675-47693
START_RANGE client25j socket2712 180 47693-47711
START_RANGE client25k socket2713 170 47711-47728
START_RANGE client25l socket2714 180 47728-47746
START_RANGE client25m socket2715 170 47746-47763
START_RANGE client25n socket2716 180 47763-47781
START_RANGE client25a socket2717 180 47781-47799
START_RANGE client25b socket2718 170 47799-47816
START_RANGE client25c socket2719 180 47816-47834
START_RANGE client25d socket2720 170 47834-47851
START_RANGE client25e socket2721 180 47851-47869
START_RANGE client25f socket2722 180 47869-47887
START_RANGE client25g socket2723 170 47887-47904
START_RANGE client25h socket2724 180 47904-47922
START_RANGE client25i socket2725 170 47922-47939
START_RANGE client25j socket2726 180 47939-47957
START_RANGE client25k socket2727 170 47957-47974
START_RANGE client25l socket2728 180 47974-47992
START_RANGE client25m socket2729 180 47992-48010
START_RANGE client25n socket2730 170 48010-48027
START_RANGE client25a socket2731 180 48027-48045
START_RANGE client25b socket2732 170 48045-48062
START_RANGE client25c socket2733 180 48062-48080
START_RANGE client25d socket2734 180 48080-48098
START_RANGE client25e socket2735 170 48098-48115
START_RANGE client25f socket2736 180 48115-48133
START_RANGE client25g socket2737 170 48133-48150
START_RANGE client25h socket2738 180 48150-48168
START_RANGE client25i socket2739 180 48168-48186
START_RANGE client25j socket2740 170 48186-48203
START_RANGE client25k socket2741 180 48203-48221
START_RANGE client25l socket2742 170 48221-48238
START_RANGE client25m socket2743 180 48238-48256
START_RANGE client25n socket2744 180 48256-48274

START_RANGE client25o socket2745 170 48274-48291
START_RANGE client25p socket2746 180 48291-48309
START_RANGE client25q socket2747 170 48309-48326
START_RANGE client25r socket2748 180 48326-48344
START_RANGE client25s socket2749 180 48344-48362
START_RANGE client25t socket2750 170 48362-48379
START_RANGE client25u socket2751 180 48379-48397
START_RANGE client25v socket2752 170 48397-48414
START_RANGE client25w socket2753 180 48414-48432
START_RANGE client25x socket2754 170 48432-48449
START_RANGE client25y socket2755 180 48449-48467
START_RANGE client25z socket2756 180 48467-48485
START_RANGE client25aa socket2757 170 48485-48502
START_RANGE client25ab socket2758 180 48502-48520
START_RANGE client25o socket2759 170 48520-48537
START_RANGE client25p socket2760 180 48537-48555
START_RANGE client25q socket2761 180 48555-48573
START_RANGE client25r socket2762 170 48573-48590
START_RANGE client25s socket2763 180 48590-48608
START_RANGE client25t socket2764 170 48608-48625
START_RANGE client25u socket2765 180 48625-48643
START_RANGE client25v socket2766 180 48643-48661
START_RANGE client25w socket2767 170 48661-48678
START_RANGE client25x socket2768 180 48678-48696
START_RANGE client25y socket2769 170 48696-48713
START_RANGE client25z socket2770 180 48713-48731
START_RANGE client25aa socket2771 180 48731-48749
START_RANGE client25ab socket2772 170 48749-48766
START_RANGE client25o socket2773 180 48766-48784
START_RANGE client25p socket2774 170 48784-48801
START_RANGE client25q socket2775 180 48801-48819
START_RANGE client25r socket2776 180 48819-48837
START_RANGE client25s socket2777 170 48837-48854
START_RANGE client25t socket2778 180 48854-48872
START_RANGE client25u socket2779 170 48872-48889
START_RANGE client25v socket2780 180 48889-48907
START_RANGE client25w socket2781 170 48907-48924
START_RANGE client25x socket2782 180 48924-48942
START_RANGE client25y socket2783 180 48942-48960
START_RANGE client25z socket2784 170 48960-48977
START_RANGE client25aa socket2785 180 48977-48995
START_RANGE client25ab socket2786 170 48995-49012
START_RANGE client25o socket2787 180 49012-49030
START_RANGE client25p socket2788 180 49030-49048
START_RANGE client25q socket2789 170 49048-49065
START_RANGE client25r socket2790 180 49065-49083
START_RANGE client25s socket2791 170 49083-49100
START_RANGE client25t socket2792 180 49100-49118
START_RANGE client25u socket2793 180 49118-49136
START_RANGE client25v socket2794 170 49136-49153
START_RANGE client25w socket2795 180 49153-49171
START_RANGE client25x socket2796 170 49171-49188
START_RANGE client25y socket2797 180 49188-49206
START_RANGE client25z socket2798 180 49206-49224
START_RANGE client25aa socket2799 170 49224-49241
START_RANGE client25ab socket2800 180 49241-49259

START_RANGE client26a socket2801 170 49259-49276
START_RANGE client26b socket2802 180 49276-49294
START_RANGE client26c socket2803 180 49294-49312
START_RANGE client26d socket2804 170 49312-49329
START_RANGE client26e socket2805 180 49329-49347
START_RANGE client26f socket2806 170 49347-49364
START_RANGE client26g socket2807 180 49364-49382
START_RANGE client26h socket2808 170 49382-49399
START_RANGE client26i socket2809 180 49399-49417
START_RANGE client26j socket2810 180 49417-49435
START_RANGE client26k socket2811 170 49435-49452
START_RANGE client26l socket2812 180 49452-49470
START_RANGE client26m socket2813 170 49470-49487
START_RANGE client26n socket2814 180 49487-49505
START_RANGE client26a socket2815 180 49505-49523
START_RANGE client26b socket2816 170 49523-49540
START_RANGE client26c socket2817 180 49540-49558
START_RANGE client26d socket2818 170 49558-49575
START_RANGE client26e socket2819 180 49575-49593
START_RANGE client26f socket2820 180 49593-49611
START_RANGE client26g socket2821 170 49611-49628
START_RANGE client26h socket2822 180 49628-49646
START_RANGE client26i socket2823 170 49646-49663
START_RANGE client26j socket2824 180 49663-49681
START_RANGE client26k socket2825 180 49681-49699
START_RANGE client26l socket2826 170 49699-49716
START_RANGE client26m socket2827 180 49716-49734
START_RANGE client26n socket2828 170 49734-49751
START_RANGE client26a socket2829 180 49751-49769
START_RANGE client26b socket2830 180 49769-49787
START_RANGE client26c socket2831 170 49787-49804
START_RANGE client26d socket2832 180 49804-49822
START_RANGE client26e socket2833 170 49822-49839
START_RANGE client26f socket2834 180 49839-49857
START_RANGE client26g socket2835 170 49857-49874
START_RANGE client26h socket2836 180 49874-49892
START_RANGE client26i socket2837 180 49892-49910
START_RANGE client26j socket2838 170 49910-49927
START_RANGE client26k socket2839 180 49927-49945
START_RANGE client26l socket2840 170 49945-49962
START_RANGE client26m socket2841 180 49962-49980
START_RANGE client26n socket2842 180 49980-49998
START_RANGE client26a socket2843 170 49998-50015
START_RANGE client26b socket2844 180 50015-50033
START_RANGE client26c socket2845 170 50033-50050
START_RANGE client26d socket2846 180 50050-50068
START_RANGE client26e socket2847 180 50068-50086
START_RANGE client26f socket2848 170 50086-50103
START_RANGE client26g socket2849 180 50103-50121
START_RANGE client26h socket2850 170 50121-50138
START_RANGE client26i socket2851 180 50138-50156
START_RANGE client26j socket2852 180 50156-50174
START_RANGE client26k socket2853 170 50174-50191
START_RANGE client26l socket2854 180 50191-50209
START_RANGE client26m socket2855 170 50209-50226
START_RANGE client26n socket2856 180 50226-50244

START_RANGE client26o socket2857 180 50244-50262
START_RANGE client26p socket2858 170 50262-50279
START_RANGE client26q socket2859 180 50279-50297
START_RANGE client26r socket2860 170 50297-50314
START_RANGE client26s socket2861 180 50314-50332

```
START_RANGE client26t socket2862 170 50332-50349
START_RANGE client26u socket2863 180 50349-50367
START_RANGE client26v socket2864 180 50367-50385
START_RANGE client26w socket2865 170 50385-50402
START_RANGE client26x socket2866 180 50402-50420
START_RANGE client26y socket2867 170 50420-50437
START_RANGE client26z socket2868 180 50437-50455
START_RANGE client26aa socket2869 180 50455-50473
START_RANGE client26ab socket2870 170 50473-50490
START_RANGE client26o socket2871 180 50490-50508
START_RANGE client26p socket2872 170 50508-50525
START_RANGE client26q socket2873 180 50525-50543
START_RANGE client26r socket2874 180 50543-50561
START_RANGE client26s socket2875 170 50561-50578
START_RANGE client26t socket2876 180 50578-50596
START_RANGE client26u socket2877 170 50596-50613
START_RANGE client26v socket2878 180 50613-50631
START_RANGE client26w socket2879 180 50631-50649
START_RANGE client26x socket2880 170 50649-50666
START_RANGE client26y socket2881 180 50666-50684
START_RANGE client26z socket2882 170 50684-50701
START_RANGE client26aa socket2883 180 50701-50719
START_RANGE client26ab socket2884 180 50719-50737
START_RANGE client26o socket2885 170 50737-50754
START_RANGE client26p socket2886 180 50754-50772
START_RANGE client26q socket2887 170 50772-50789
START_RANGE client26r socket2888 180 50789-50807
START_RANGE client26s socket2889 170 50807-50824
START_RANGE client26t socket2890 180 50824-50842
START_RANGE client26u socket2891 180 50842-50860
START_RANGE client26v socket2892 170 50860-50877
START_RANGE client26w socket2893 180 50877-50895
START_RANGE client26x socket2894 170 50895-50912
START_RANGE client26y socket2895 180 50912-50930
START_RANGE client26z socket2896 180 50930-50948
START_RANGE client26aa socket2897 170 50948-50965
START_RANGE client26ab socket2898 180 50965-50983
START_RANGE client26o socket2899 170 50983-51000
START_RANGE client26p socket2900 180 51000-51018
START_RANGE client26q socket2901 180 51018-51036
START_RANGE client26r socket2902 170 51036-51053
START_RANGE client26s socket2903 180 51053-51071
START_RANGE client26t socket2904 170 51071-51088
START_RANGE client26u socket2905 180 51088-51106
START_RANGE client26v socket2906 180 51106-51124
START_RANGE client26w socket2907 170 51124-51141
START_RANGE client26x socket2908 180 51141-51159
START_RANGE client26y socket2909 170 51159-51176
START_RANGE client26z socket2910 180 51176-51194
START_RANGE client26aa socket2911 180 51194-51212
START_RANGE client26ab socket2912 170 51212-51229

START_RANGE client27a socket2913 180 51229-51247
START_RANGE client27b socket2914 170 51247-51264
START_RANGE client27c socket2915 180 51264-51282
START_RANGE client27d socket2916 170 51282-51299
START_RANGE client27e socket2917 180 51299-51317
START_RANGE client27f socket2918 180 51317-51335
START_RANGE client27g socket2919 170 51335-51352
START_RANGE client27h socket2920 180 51352-51370
START_RANGE client27i socket2921 170 51370-51387
START_RANGE client27j socket2922 180 51387-51405
START_RANGE client27k socket2923 180 51405-51423
START_RANGE client27l socket2924 170 51423-51440
START_RANGE client27m socket2925 180 51440-51458
START_RANGE client27n socket2926 170 51458-51475
START_RANGE client27a socket2927 180 51475-51493
START_RANGE client27b socket2928 180 51493-51511
START_RANGE client27c socket2929 170 51511-51528
START_RANGE client27d socket2930 180 51528-51546
START_RANGE client27e socket2931 170 51546-51563
START_RANGE client27f socket2932 180 51563-51581
START_RANGE client27g socket2933 180 51581-51599
START_RANGE client27h socket2934 170 51599-51616
START_RANGE client27i socket2935 180 51616-51634
START_RANGE client27j socket2936 170 51634-51651
START_RANGE client27k socket2937 180 51651-51669
START_RANGE client27l socket2938 180 51669-51687
START_RANGE client27m socket2939 170 51687-51704
START_RANGE client27n socket2940 180 51704-51722
START_RANGE client27a socket2941 170 51722-51739
START_RANGE client27b socket2942 180 51739-51757
START_RANGE client27c socket2943 170 51757-51774
START_RANGE client27d socket2944 180 51774-51792
START_RANGE client27e socket2945 180 51792-51810
START_RANGE client27f socket2946 170 51810-51827
START_RANGE client27g socket2947 180 51827-51845
START_RANGE client27h socket2948 170 51845-51862
START_RANGE client27i socket2949 180 51862-51880
START_RANGE client27j socket2950 180 51880-51898
START_RANGE client27k socket2951 170 51898-51915
START_RANGE client27l socket2952 180 51915-51933
START_RANGE client27m socket2953 170 51933-51950
START_RANGE client27n socket2954 180 51950-51968
START_RANGE client27a socket2955 180 51968-51986
START_RANGE client27b socket2956 170 51986-52003
START_RANGE client27c socket2957 180 52003-52021
START_RANGE client27d socket2958 170 52021-52038
START_RANGE client27e socket2959 180 52038-52056
START_RANGE client27f socket2960 180 52056-52074
START_RANGE client27g socket2961 170 52074-52091
START_RANGE client27h socket2962 180 52091-52109
START_RANGE client27i socket2963 170 52109-52126
START_RANGE client27j socket2964 180 52126-52144
START_RANGE client27k socket2965 180 52144-52162
START_RANGE client27l socket2966 170 52162-52179
START_RANGE client27m socket2967 180 52179-52197
START_RANGE client27n socket2968 170 52197-52214

START_RANGE client27o socket2969 180 52214-52232
START_RANGE client27p socket2970 170 52232-52249
START_RANGE client27q socket2971 180 52249-52267
START_RANGE client27r socket2972 180 52267-52285
START_RANGE client27s socket2973 170 52285-52302
START_RANGE client27t socket2974 180 52302-52320
START_RANGE client27u socket2975 170 52320-52337
START_RANGE client27v socket2976 180 52337-52355
START_RANGE client27w socket2977 180 52355-52373
START_RANGE client27x socket2978 170 52373-52390
START_RANGE client27y socket2979 180 52390-52408
START_RANGE client27z socket2980 170 52408-52425
START_RANGE client27aa socket2981 180 52425-52443
START_RANGE client27ab socket2982 180 52443-52461
START_RANGE client27o socket2983 170 52461-52478
START_RANGE client27p socket2984 180 52478-52496
START_RANGE client27q socket2985 170 52496-52513
START_RANGE client27r socket2986 180 52513-52531
START_RANGE client27s socket2987 180 52531-52549
START_RANGE client27t socket2988 170 52549-52566
START_RANGE client27u socket2989 180 52566-52584
START_RANGE client27v socket2990 170 52584-52601
START_RANGE client27w socket2991 180 52601-52619
START_RANGE client27x socket2992 180 52619-52637
START_RANGE client27y socket2993 170 52637-52654
START_RANGE client27z socket2994 180 52654-52672
START_RANGE client27aa socket2995 170 52672-52689
START_RANGE client27ab socket2996 180 52689-52707
START_RANGE client27o socket2997 170 52707-52724
START_RANGE client27p socket2998 180 52724-52742
START_RANGE client27q socket2999 180 52742-52760
START_RANGE client27r socket3000 170 52760-52777
START_RANGE client27s socket3001 180 52777-52795
START_RANGE client27t socket3002 170 52795-52812
START_RANGE client27u socket3003 180 52812-52830
START_RANGE client27v socket3004 180 52830-52848
START_RANGE client27w socket3005 170 52848-52865
START_RANGE client27x socket3006 180 52865-52883
START_RANGE client27y socket3007 170 52883-52900
START_RANGE client27z socket3008 180 52900-52918
START_RANGE client27aa socket3009 180 52918-52936
START_RANGE client27ab socket3010 170 52936-52953
START_RANGE client27o socket3011 180 52953-52971
START_RANGE client27p socket3012 170 52971-52988
START_RANGE client27q socket3013 180 52988-53006
START_RANGE client27r socket3014 180 53006-53024
START_RANGE client27s socket3015 170 53024-53041
START_RANGE client27t socket3016 180 53041-53059
START_RANGE client27u socket3017 170 53059-53076
START_RANGE client27v socket3018 180 53076-53094
START_RANGE client27w socket3019 180 53094-53112
START_RANGE client27x socket3020 170 53112-53129
START_RANGE client27y socket3021 180 53129-53147
START_RANGE client27z socket3022 170 53147-53164
START_RANGE client27aa socket3023 180 53164-53182
START_RANGE client27ab socket3024 180 53182-53200

#elif MASTER_NUM10
#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(int length, 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's inout.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 the max 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 die flag\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 Warehouse prompt\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 logging in\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 Vendors

The eCommerce Transaction Platform

May 2, 2003

Margie Blevins
Project Manager, AIX Performance
IBM
11501 Burnet Road
Austin, TX 78759
(512) 838-3176 Office
(512) 838-1852 FAX

Per your request I am enclosing the pricing information regarding TUXEDO 8.0 that you requested. This pricing applies to Tuxedo 6.4, 6.5, 7.1, 8.0 and 8.1. Please note that Tuxedo 8.1 is our most recent version of Tuxedo available. Core Functionality Services (CFS)–R pricing is appropriate for your activities. As per the table below IBM systems are classified as either a Tier 1, 2, 3, 4 or 5 systems depending on the performance and CPU capacity of the system. For example an IBM p630 series system would be a Tier 3 machine. The extended list license charge for 27 systems would be: 27 servers * 12,000 = $324,000. This would qualify for a 15% license discount. Additionally, support charges are discountable by 15%.

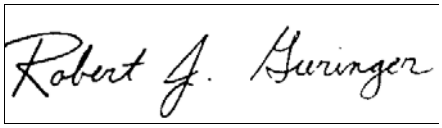
This quote is valid for 90 days from the date of this letter.

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

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

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

Very Truly Yours,



Robert J. Gieringer

Worldwide Pricing Manager

05/02/2003 BEA Systems, Inc.

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

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

	Tier 1	Tier 1	Tier 2	Tier 3	Tier 3	Tier 4	Tier 5
Platform							
IBM RS6000	M20 250 25S 40P 43P 25T 25W E20	34H 355 C10 C20 41T 41W 42T 42W	44P-170 43P- 140, 150 B-50,F3L 365 375 39H 41X 590, 595 E30,F30,F40 PSeries 610	44P-270 52X 530X R10 H10 550L 580 3AT,3BT pSeries 660 - 6H0/6H1/6M1 pSeries 620 6F0/6F1 pSeries 630 pSeries 640 pSeries 650	58H 980B,990 59H F50,F80,M80 G30, G40 H50, H70,H80 R20, R24,R30,R40,R50 J30, J40, J50	S70,S7A S80, Pseries 680 (S85) Pseries 670	Pseries - 690
IBM RS/6000 SP Server (SP2)							RS/6000 SP Server

http://www.netgear.com/products/prod_details.asp?prodID=22&view=

NETGEAR

Powered by

Netgear 10/100 16 Port Dual Speed Switch with Uplink Button

Model: FS116

Dealer	Dealer Product Description	Price	In Stock	
amazon.com	Netgear FS116 10/100 16-Port Dual Speed Switch with Uplink Button	$105.99	Yes	Buy
buy.com	Netgear Features FS116NA	$97.99	Yes	Buy
COMPUTERS 4sure.com	Netgear FS116 10/100 16-Port Switch FS116NA	$101.95	Yes	Buy
PC Connection	Netgear Inc., FS116 10/100BaseTX 16-Port Fast Ethernet Switch + Uplink	$109.95	Yes	Buy
Gateway	Netgear FS116 16-PORT 10/100BASE-TX SWITCH FS116NA	$107.95	Yes	Buy
Insight	Netgear FS116 - Switch - 16 port(s) - 10Base-T, 100Base-TX - 100 Mbps - EN, Fast EN FS116NA	$105.99	Yes	Buy
RCMall	Netgear FS116 16 port 10/100 switch	$99.99	Yes	Buy
CDW	Netgear FS116 FS116NA	$117.22	Yes	Buy
MICRO WAREHOUSE	Netgear FS116 16-Port 10/100 Switch W/Uplink FS116NA	$109.95	Yes	Buy
ZONES	Netgear FS116 - Switch FS116NA	$105.98	Yes	Buy

Appendix G: Pricing Information

The price of the IBM-supplied configuration will be available for at least sixty day. Prices are in US Dollars and are for sale within he 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:

IBM Hardware:

Bill Casey
eServer pSeries Offering Manager
wrcasey@us.ibm.com
(512) 838-1422

IBM DB2:

Paul Rivot
Director, Database Servers and Business Intelligence Software
privot@us.ibm.com
(914) 766-1325

Thank you,

Robert Schramer,
Mgr, Product Pricing & Business Evaluation - eServer pSeries
schramer@us.ibm.com
(512) 838-5297