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

Second Edition

September 10, 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

Second Edition September 10, 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:

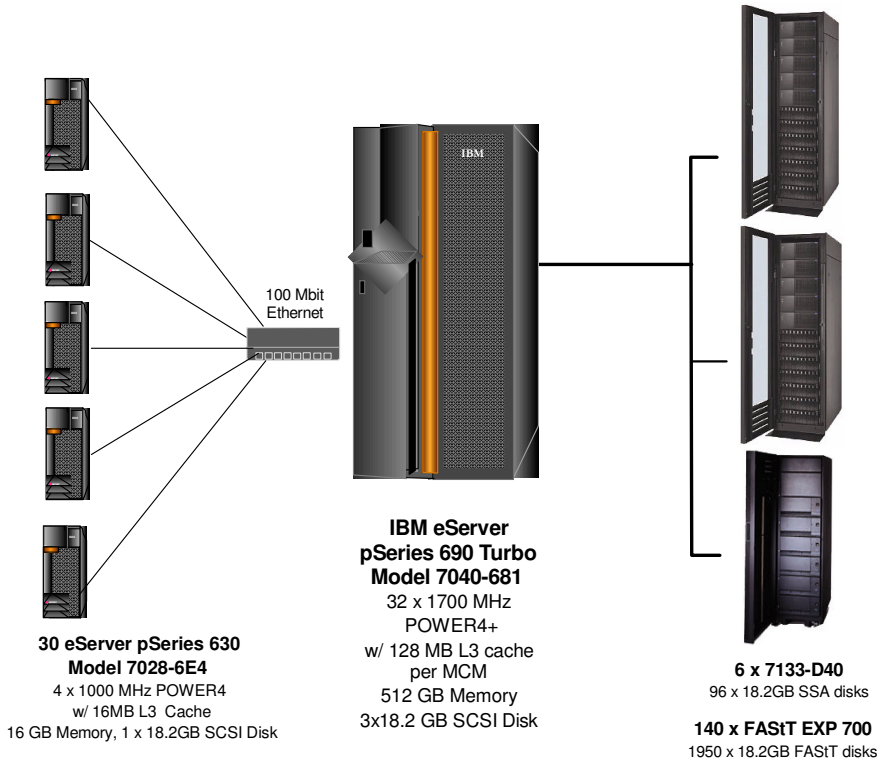
TPC Benchmark Administrator
IBM Commercial Performance
Mail Stop 9571
11501 Burnet Road
Austin, TX 78758
FAX Number (512) 838-1852

© 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 eServer pSeries 690 Turbo Model 7040-681		TPC-C Rev. 5.1 Report Date: June 30, 2003	
Total System Cost	TPC-C Throughput	Price/Performance	Availability Date	
$6,301,741	763,898.39 tpmC	$8.25/tpmC	November 8, 2003	
Processors	Database Manager	Operating System	Other Software	No. Users
32 for database 120 for clients	DB2 UDB 8.1	AIX 5L V5.2	TUXEDO 8.0	609,000



System Components	Clients		Server	
	Quantity	Description	Quantity	Description
Processor	30	4 x1000 MHz POWER4 W/16MB L3 cache each	1	32 x 1700 MHz POWER4+ w/ 128 MB L3 cache per MCM, 4 MCM in the SUT
Memory	16 GB		512 GB	
Disk Controllers	2	Ultra3 SCSI-2 (Integrated)	1 2 26	SCSI-2 (Integrated) SSA Adapters Fibre Channel Adapters
Disk Drives	1	18.2 GB Ultra SCSI each client	3 96 1950	18.2 GB SCSI disk 18.2 GB SSA disk 18 .2 GB FAStT disk
Total Storage	16.96 GB each client		33,300.03GB	
Terminals	1	System Console	1	System Console

Server Hardware	Part No.	Source	Unit Price	Qty	Ext Price	Maint Price
IBM eServer pSeries 690 (CD, cables, clock card)	7040-681	1	13,394	1	13,394	20,760
128MB (4x32) L3 Cache, 1700MHz 8w MCM	4199	1	375,000	4	1,500,000	372,000
64GB H-Memory (inward and outward 4488,4489)	4488	1	344,000	4	1,376,000	0
Std CEC Fan Pwr cbl grp, service proc. 6161,6162	6161	1	600	1	600	0
Standard CEC DC/DC Converter Assembly	6170	1	4,700	2	9,400	0
Power Cable Group, 1 thru 4 proc. module	6181	1	720	1	720	0
Additional DC/DC Converter Assembly, (DCA)	6189	1	4,200	4	16,800	0
Capacitor book (for 2 MCMs)	6198	1	1,800	2	3,600	0
Service Proc. + Dual Rio Loop (GX), I/O loop adapter	6418	1	15,000	1	15,000	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
Converter Cable, VHDCI to P, 0.3M	2118	1	60	4	240	0
SCSI cable - B&C to Media Drawer	2122	1	275	1	275	0
Fibre Chan. Converter Cable, LC (M) to SC (F), 2M	2146	1	100	4	400	0
RIO-G Cable, 500MHz, 1.2m, 2*1.75m (3146,3156)	3146	1	1250	4	5,000	0
18.2 GB Ultra3 10K RPM 1" U3 Module	3158	1	1,540	3	4,620	0
10/100 Mbps Ethernet adapter II	4962	1	275	10	2,750	0
Cable Grp, 4xUPIC/2xRIO, BPM(IO#1 thru 3) (6121-6123)	6121	1	400	3	1,200	0
I/O Drwr DC/DC Converter assembly, (DCA)	6172	1	4000	8	32,000	0
Power Cable, B&C to Media Drwr 6179, SCSI adapter 620	6179	1	1,090	1	1,090	0
Adv. SerialRAID Adapter, 32MB Cache (6230, 6235)	6230	1	3575	2	7,150	0
2 Gigabit Fibre Channel PCI-X Adapter	6239	1	3,100	26	80,600	0
Ultra3 SCSI 4-Pack	6564	1	500	16	8,000	0
I/O Drawer PCI-X Planar, 10 Slot, 2 Integrated	6571	1	8,000	8	64,000	0
Rack, Front door, back door	7040-61R	1	9,250	1	9,250	3,960
Bulk Power Regulator (BPR)	6186	1	4000	6	24,000	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
IBM T541H 15 TFT Color Monitor, keyboard, graphics	3637	1	974	1	974	0
				Subtotal	**3,273,958**	**413,040**
FASt Storage (Data Disks)						
RS/6000 System Rack (6081,6098)	7014-T42	1	4,270	13	55,510	11,544
FAStT EXP 700	1740-1RU	1	6,000	140	840,000	201,600
Short Wave SFP	2210	1	499	560	279,440	0
18.2 GB 15K drive	5211	1	681	1,950	1,327,950	0
FAStT900 Storage Server	1742-90U	1	66,500	13	864,500	26,780
Short Wave SFP GBIC	2210	1	499	104	51,896	0
FAStT700 Mini HUB	3507	1	899	26	23,374	0
				Subtotal	**3,442,670**	**239,924**

SSA Storage (Log Disks)						
System Rack Model T42	7014-T42	1	4,270	1	4,270	888
Additional Power Distribution Unit	6171	1	1,000	1	1,000	0
				Subtotal	**5,270**	**888**
SSA Disk Subsys, 50/60 Hz AC, pwr supply,cover,cables	7133-D40	1	15,125	6	90,750	26,496
Advanced 10K rpm/18.2 GB Disk Module	8536	1	4,500	96	432,000	0
				Subtotal	**522,750**	**26,496**
Server Software						
AIX 5.2 (media only)	5962-A5L	1	50	1	50	0
AIX SWMA Subsription	5771-SUB	1	247	96	23,712	0
AIX SWMA Support	5771-SPT	1	697	96	0	66,912
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	**656,437**	**127,278**
Client Hardware and Software						
pSeries 630-6E4, CD-ROM, 18.2GB Disk, Pwr Supply	7028-6E4	1	3,941	30	118,230	90,000
FC I Exp Config (402 E) AIX	8016	1	39,249	30	1,177,470	
T541H 15" TFT Color Monitor, Captured Cable, keyboard	3637	1	562	1	562	0
POWER GXT135P Graphics Accelerator -Digital support	2849	1	412	1	412	0
SCSI Connector Cable + 4-Pack Hot Swap Back plane	4254	1	354	30	10,620	0
4-port 10/100 Mbps Ethernet adapter	4961	1	1,250	90	112,500	0
AIX SWMA Subsription	5771-SUB	1	157	240	37,680	0
AIX SWMA Support	5771-SPT	1	427	240	0	102,480
AIX 5L version 5	5962-A5L	1	50	30	1,500	0
				Subtotal	**1,458,974**	**192,480**
Third Party Hardware/Software						
BEA Tuxedo 8.0		2	10,200	30	306,000	192,780
10/100 16-Port Dual Speed Switch	FS116NA	3	81	5	405	0
				Subtotal	**306,405**	**192,780**
				Discounts	**(4,167,191)**	**(390,418)**
"price updated on August 12, 2003 to reflect accurate software maintenance charges"				**Total**	**5,499,273**	**802,468**
Audited by: Francois Raab, Info Sizing (www.infosizing.com)						
Pricing Sources:					**Three-Year Cost of Ownership**	**$6,301,741**
1=IBM HW: Bill Casey, eServer pSeries Offering Manager, wrcasey@us.ibm.com, 512-838-1422					**tpmC**	**763,898.39**
1= IBM DB2: Paul Rivot, Director Database Servers and Business Intelligence Software,					**$/tpmC**	**$8.25**
e-mail privot@us.ibm.com, phone 1-914-766-1325						
2= BEA Systems						
3=NETGEAR						

Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you

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

MQTH, computed Maximum Qualified Throughput: 763,898.39 ttpmC

Response Times (in seconds)	90th %	Average	Maximum
New Order	1.40	0.58	20.39
Payment	1.14	0.40	17.41
Order-Status	1.18	0.44	14.44
Delivery (interactive)	1.12	0.38	9.60
Delivery (deferred)	2.26	1.04	9.25
Stock-Level	1.14	0.40	10.60
Menu	0.00	0.00	15.01

Transaction Mix, in percent of total transactions	Percent
New Order	44.90%
Payment	43.03%
Order-Status	4.01%
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	18.09/120.20
Payment	3.00/0.01	3.01/12.02	3.09/120.20
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	2.07/50.20

Test Duration	
Ramp-up Time	31 minutes 00 seconds
Measurement interval	2 hours 10 minutes
Transactions during measurement interval (all types)	221,132,757
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
$6,301,741	763,898.39	$8.25 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.*

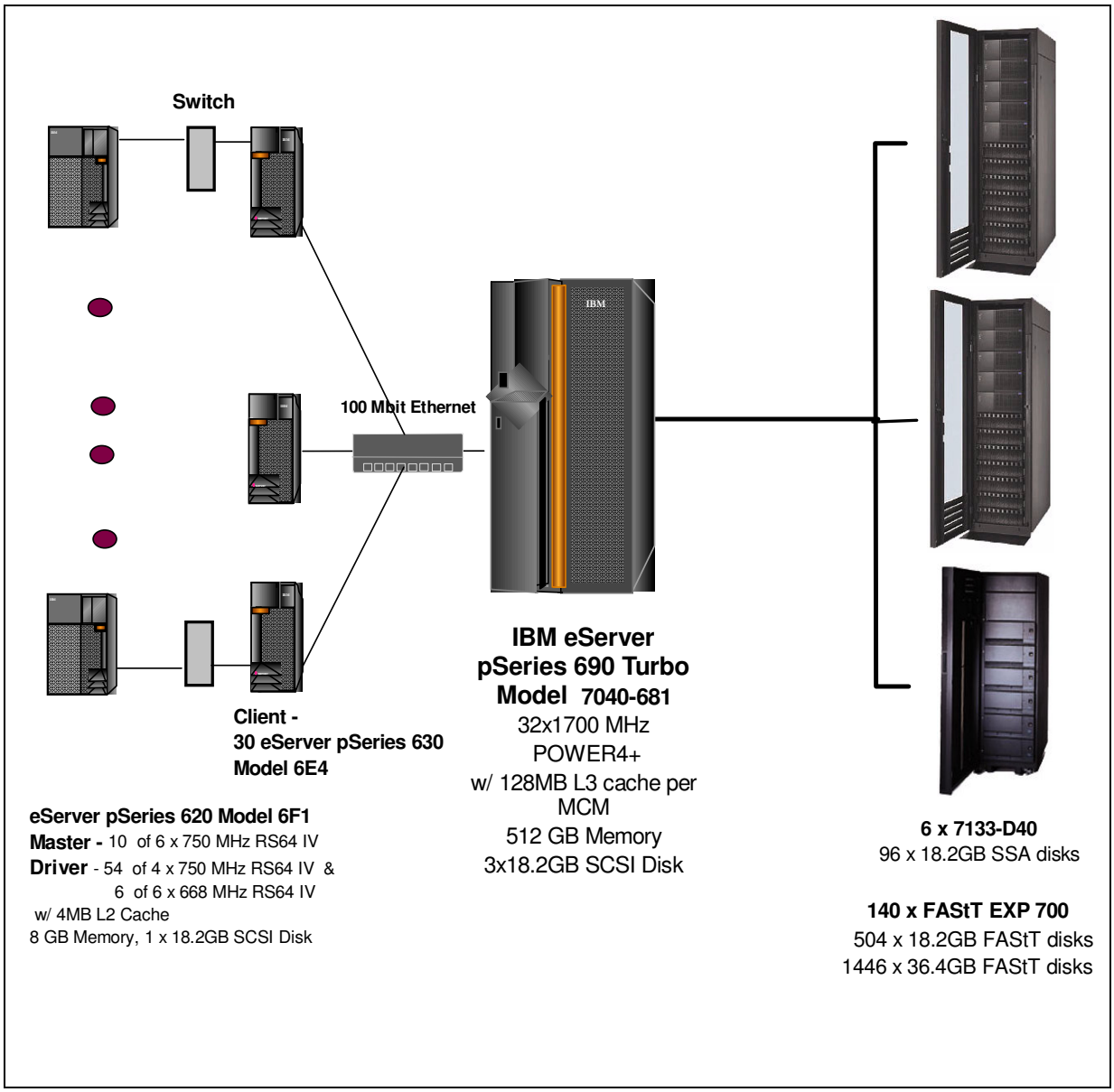
Appendix B contains the system, data base, and application parameters changed from their default values used in these TPC Benchmark™ C tests.

1.4 Configuration Diagrams

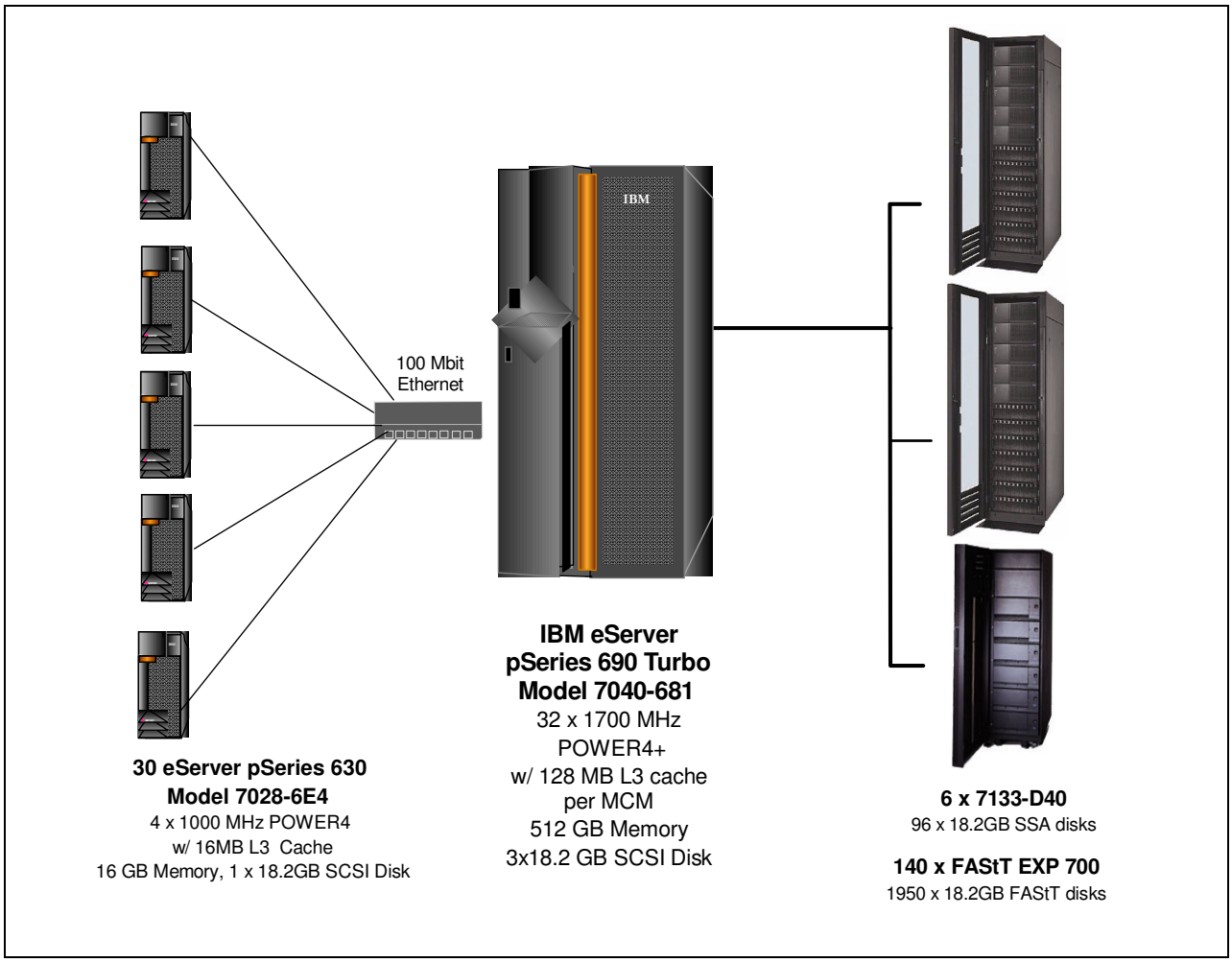
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- *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:

- On the Clients, one disk was installed but not used. It has no filesystems or logical volumes defined on it.
- The 1446 36.4GB disks attached to the Server were substitued with 18.2GB disks.

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 1740 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	99.01%
Percentage of Remote order lines	0.99%
Percentage of Rolled Back Transactions	0.99%
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	60.00%
Percentage of Order-Status using C_LAST	59.98%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.90%
Payment	43.03%
Order-Status	4.01%
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, failure of Durable Fast Write Cache on SSA Adapter for Redo Logs, Instantaneous Interruption, and Memory Failure

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

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving SUM_1.
2. A full load test was started and allowed to run for over 7 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 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.

6. The run continued without the cache.
7. The system was subsequently shut down.
8. The disk from step 3 was replaced, and the fast write cache was enabled.
9. The system was powered back on and DB2 allowed to recover.
10. Step 1 is performed returning the value for SUM_2. It was verified that SUM_2 - SUM_1 was greater than the number of completed New_Order transactions recorded by the RTE.
11. Consistency condition 3 was verified.

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, failure of a disk controller, and failure of the disk controller's cache battery.

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 approximately 17% of the full load.
4. Even though fast write cache was not enabled, the cache battery was removed. The run continued without interruption
5. Three minutes later, a controller was removed and the run continued.
6. A disk containing the TPCC table was removed.
7. The run was left to complete, recording many errors due to the disk failure.
8. The cache controller, cache battery, and disk from step 6 were powered back on and the full database was restored from the backup copy in step 1.
9. DB2 was restarted and its transaction log was used to roll forward through the transactions that had completed since the run had started.
10. Step 2 was performed returning SUM_2. It was verified that SUM_2 - SUM_1 was equal to the number of completed New_Order transactions recorded by the RTE.
11. 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).

The Customer, OrderLine, and Stock tables are based on 60,900 warehouses, the number of active warehouses during the benchmark. Since these are partitioned tables, data for the unused warehouses were easily eliminated. The remaining tables are based on 67,860 warehouses, the number of warehouses used to initially populate the database.

Table Name	Number of Rows
Warehouse	60,900
District	678,600
Customer	1,827,000,000
History	2,035,800,000
Order	2,035,800,000
New Order	610,740,000
Order Line	18,269,972,420
Stock	6,090,000,000
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.

There are 2 storage adapters for the log. Each adapter has 6 logical disks (hdisks) and each hdisk has 8 physical disks, for a total of 96 log disks. The hdisks are configured as RAID5 disk arrays. Each physical disk has a capacity of 18GB. The log LV is striped across the 12 hdisks.

There are 26 storage adapters for the data disks. Each adapter has 3 hdisks and each hdisk has 25 physical disks, for a total of 1,950 data disks. Each hdisk is configured as a RAID0 storage array. The physical disks are a mixture of 18GB and 36GB drives and every hdisk has at least one 18GB drive. The FAStT RAID software caps the capcity of the 36GB drives to 18GB, the lowest capacity drive in the array. Since there was no added capacity or performance, as demonstrated by disk stress testing, the 36 GB drives are substitued with 18GB drives for pricing.

Each data logical volume (LV) is striped across 2 hdisks. The 2 hdisks are on separate adapters. There are a total of 36 LVs per adapter pair.

The following convention is used for the LV names: D1FaaVbttt##, where

aa = disk storage rack number
b = Volume Group (VG)
ttt= table name
= the partition number of the table.

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

Disk Mapping:

hdisk	Adapter	Logical Volumes
hdisk38	fcs3	D1F01V1ITEM D1F01V1WARE D1F01V1DIST D1F01V1NORI D1F01V1CSTI01 D1F01V1ORL01 D1F01V1STK01 D1F01V1CST01 D1F01V1ORDI D1F01V1ORD D1F01V1NOR D1F01V1HIST
hdisk39	fcs2	
hdisk40	fcs3	D1F12V2ITEM D1F01V2WARE D1F01V2DIST D1F01V2NORI D1F01V2CSTI02 D1F01V2ORL02 D1F01V2STK02 D1F01V2CST02 D1F01V2ORDI D1F01V2ORD D1F01V2NOR D1F01V2HIST
hdisk41	fcs2	
hdisk42	fcs3	D1F01V3ITEM D1F01V3WARE D1F01V3DIST D1F01V3NORI D1F01V3CSTI03 D1F01V3ORL03 D1F01V3STK03 D1F01V3CST03 D1F01V3ORDI D1F01V3ORD D1F01V3NOR D1F01V3HIST
hdisk43	fcs2	
hdisk44	fcs6	D1F02V1ITEM D1F02V1WARE D1F02V1DIST D1F02V1NORI D1F02V1CSTI04 D1F02V1ORL04 D1F02V1STK04 D1F02V1CST04 D1F02V1ORDI D1F02V1ORD D1F02V1NOR D1F02V1HIST
hdisk45	fcs5	
hdisk46	fcs6	D1F02V2ITEM D1F02V2WARE D1F02V2DIST D1F02V2NORI D1F02V2CSTI05 D1F02V2ORL05 D1F02V2STK05 D1F02V2CST05 D1F02V2ORDI D1F02V2ORD D1F02V2NOR D1F02V2HIST
hdisk47	fcs5	
hdisk48	fcs6	D1F02V3ITEM D1F02V3WARE D1F02V3DIST D1F02V3NORI D1F02V3CSTI36 D1F02V3ORL06 D1F03V3STK06 D1F03V3CST06 D1F03V3ORDI D1F03V3ORD D1F03V3NOR D1F03V3HIST
hdisk49	fcs5	
hdisk70	fcs17	D1F03V1ITEM D1F03V1WARE D1F03V1DIST D1F03V1NORI D1F03V1CSTI07 D1F03V1ORL07 D1F03V1STK07 D1F03V1CST07 D1F03V1ORDI D1F03V1ORD D1F03V1NOR D1F03V1HIST
hdisk71	fcs16	
hdisk72	fcs17	D1F03V2ITEM D1F03V2WARE D1F03V2DIST D1F03V2NORI D1F03V2CSTI08 D1F03V2ORL08 D1F03V2STK08 D1F03V2CST08 D1F03V2ORDI D1F03V2ORD D1F03V2NOR D1F03V2HIST
hdisk73	fcs16	
hdisk74	fcs17	D1F03V3ITEM D1F03V3WARE D1F03V3DIST D1F03V3NORI D1F03V3CSTI09 D1F03V3ORL09 D1F03V3STK09 D1F03V3CST09 D1F03V3ORDI D1F03V3ORD D1F03V3NOR D1F03V3HIST
hdisk75	fcs18	
hdisk77	fcs22	D1F04V1ITEM D1F04V1WARE D1F04V1DIST D1F04V1NORI D1F04V1CSTI10 D1F04V1ORL10 D1F04V1STK10 D1F04V1CST10 D1F04V1ORDI D1F04V1ORD D1F04V1NOR D1F04V1HIST
hdisk78	fcs21	
hdisk79	fcs22	D1F04V2ITEM D1F04V2WARE D1F04V2DIST D1F04V2NORI D1F04V2CSTI11 D1F04V2ORL11 D1F04V2STK11 D1F04V2CST11 D1F04V2ORDI D1F04V2ORD D1F04V2NOR D1F04V2HIST
hdisk80	fcs21	
hdisk81	fcs22	D1F04V3ITEM D1F04V3WARE D1F04V3DIST D1F04V3NORI D1F04V3CSTI12 D1F04V3ORL12 D1F04V3STK12 D1F04V3CST12 D1F04V3ORDI D1F04V3ORD D1F04V3NOR D1F04V3HIST
hdisk82	fcs21	
hdisk83	fcs24	D1F05V1ITEM D1F05V1WARE D1F05V1DIST D1F05V1NORI D1F05V1CSTI13 D1F05V1ORL13 D1F05V1STK13 D1F05V1CST13 D1F05V1ORDI D1F05V1ORD D1F05V1NOR D1F05V1HIST
hdisk84	fcs23	
hdisk85	fcs24	D1F05V2ITEM D1F05V2WARE D1F05V2DIST D1F05V2NORI D1F05V2CSTI14 D1F05V2ORL14 D1F05V2STK14 D1F05V2CST14 D1F05V2ORDI D1F05V2ORD D1F05V2NOR D1F05V2HIST
hdisk86	fcs23	
hdisk87	fcs24	D1F05V3ITEM D1F05V3WARE D1F05V3DIST D1F05V3NORI D1F05V3CSTI15 D1F05V3ORL15 D1F05V3STK15 D1F05V3CST15 D1F05V3ORDI D1F05V3ORD D1F05V3NOR D1F05V3HIST
hdisk88	fcs23	
hdisk89	fcs26	D1F06V1ITEM D1F06V1WARE D1F06V1DIST D1F06V1NORI D1F06V1CSTI16 D1F06V1ORL16 D1F06V1STK16 D1F06V1CST16 D1F06V1ORDI D1F06V1ORD D1F06V1NOR D1F06V1HIST
hdisk90	fcs25	
hdisk91	fcs26	D1F06V2ITEM D1F06V2WARE D1F06V2DIST D1F06V2NORI D1F06V2CSTI17 D1F06V2ORL17 D1F06V2STK17 D1F06V2CST17 D1F06V2ORDI D1F06V2ORD D1F06V2NOR D1F06V2HIST
hdisk92	fcs25	
hdisk93	fcs26	D1F06V3ITEM D1F06V3WARE D1F06V3DIST D1F06V3NORI D1F06V3CSTI18 D1F06V3ORL18 D1F06V3STK18 D1F06V3CST18 D1F06V3ORDI D1F06V3ORD D1F06V3NOR D1F05V3HIST
hdisk94	fcs25	
hdisk95	fcs28	D1F07V1ITEM D1F07V1WARE D1F07V1DIST D1F07V1NORI D1F07V1CSTI19 D1F07V1ORL19 D1F07V1STK19 D1F07V1CST19 D1F07V1ORDI D1F07V1ORD D1F07V1NOR D1F07V1HIST
hdisk96	fcs27	
hdisk97	fcs28	D1F07V2ITEM D1F07V2WARE D1F07V2DIST D1F07V2NORI D1F07V2CSTI20 D1F07V2ORL20 D1F07V2STK20 D1F07V2CST20 D1F07V2ORDI D1F07V2ORD D1F07V2NOR D1F07V2HIST
hdisk98	fcs27	
hdisk99	fcs28	D1F07V3ITEM D1F07V3WARE D1F07V3DIST D1F07V3NORI D1F07V3CSTI21 D1F07V3ORL21 D1F07V3STK21 D1F07V3CST21 D1F07V3ORDI D1F07V3ORD D1F07V3NOR D1F05V3HIST
hdisk100	fcs27	
hdisk101	fcs29	D1F08V1ITEM D1F08V1WARE D1F08V1DIST D1F08V1NORI D1F08V1CSTI22 D1F08V1ORL22 D1F08V1STK22 D1F08V1CST22 D1F08V1ORDI D1F08V1ORD D1F08V1NOR D1F08V1HIST
hdisk102	fcs30	
hdisk103	fcs29	D1F08V2ITEM D1F08V2WARE D1F08V2DIST D1F08V2NORI D1F08V2CSTI23 D1F08V2ORL23 D1F08V2STK23 D1F08V2CST23 D1F08V2ORDI D1F08V2ORD D1F08V2NOR D1F08V2HIST
hdisk104	fcs30	
hdisk105	fcs29	D1F08V3ITEM D1F08V3WARE D1F08V3DIST D1F08V3NORI D1F08V3CSTI24 D1F08V3ORL24 D1F08V3STK24 D1F08V3CST24 D1F08V3ORDI D1F08V3ORD D1F08V3NOR D1F05V3HIST
hdisk106	fcs30	
hdisk107	fcs32	D1F09V1ITEM D1F09V1WARE D1F09V1DIST D1F09V1NORI D1F09V1CSTI25 D1F09V1ORL25 D1F09V1STK25 D1F09V1CST25 D1F09V1ORDI D1F09V1ORD D1F09V1NOR D1F09V1HIST
hdisk108	fcs31	
hdisk109	fcs32	D1F09V2ITEM D1F09V2WARE D1F09V2DIST D1F09V2NORI D1F09V2CSTI26 D1F09V2ORL26 D1F09V2STK26 D1F09V2CST26 D1F09V2ORDI D1F09V2ORD D1F09V2NOR D1F09V2HIST
hdisk110	fcs31	

hdisk111	fcs32	D1F09V3ITEM D1F09V3WARE D1F09V3DIST D1F09V3NORI D1F09V3CSTI27 D1F09V3ORL27 D1F09V3STK27 D1F09V3CST27 D1F09V3ORDI D1F09V3ORD D1F09V3NOR D1F05V3HIST
hdisk112	fcs31	
hdisk113	fcs34	D1F10V1ITEM D1F10V1WARE D1F10V1DIST D1F10V1NORI D1F10V1CSTI28 D1F10V1ORL28 D1F10V1STK28 D1F10V1CST28 D1F10V1ORDI D1F10V1ORD D1F10V1NOR D1F10V1HIST
hdisk114	fcs33	
hdisk115	fcs34	D1F10V2ITEM D1F10V2WARE D1F10V2DIST D1F10V2NORI D1F10V2CSTI30 D1F10V2ORL29 D1F10V2STK29 D1F10V2CST29 D1F10V2ORDI D1F10V2ORD D1F10V2NOR D1F10V2HIST
hdisk116	fcs33	
hdisk117	fcs34	D1F10V3ITEM D1F10V3WARE D1F10V3DIST D1F10V3NORI D1F10V3CSTI30 D1F10V3ORL30 D1F10V3STK30 D1F10V3CST30 D1F10V3ORDI D1F10V3ORD D1F10V3NOR D1F05V3HIST
hdisk118	fcs33	
hdisk119	fcs36	D1F11V1ITEM D1F11V1WARE D1F11V1DIST D1F11V1NORI D1F11V1CSTI31 D1F11V1ORL31 D1F11V1STK31 D1F11V1CST31 D1F11V1ORDI D1F11V1ORD D1F11V1NOR D1F11V1HIST
hdisk120	fcs35	
hdisk121	fcs36	D1F11V2ITEM D1F11V2WARE D1F11V2DIST D1F11V2NORI D1F11V2CSTI32 D1F11V2ORL32 D1F11V2STK32 D1F11V2CST32 D1F11V2ORDI D1F11V2ORD D1F11V2NOR D1F11V2HIST
hdisk122	fcs35	
hdisk123	fcs36	D1F11V3ITEM D1F11V3WARE D1F11V3DIST D1F11V3NORI D1F11V3CSTI33 D1F11V3ORL33 D1F11V3STK33 D1F11V3CST33 D1F11V3ORDI D1F11V3ORD D1F11V3NOR D1F05V3HIST
hdisk124	fcs35	
hdisk32	fcs1	D1F12V1ITEM D1F12V1WARE D1F12V1DIST D1F12V1NORI D1F12V1CSTI34 D1F12V1ORL34 D1F12V1STK34 D1F12V1CST34 D1F12V1ORDI D1F12V1ORD D1F01V1NOR D1F01V1HIST
hdisk33	fcs0	
hdisk34	fcs1	D1F12V1ITEM D1F12V1WARE D1F12V1DIST D1F12V1NORI D1F12V1CSTI35 D1F12V1ORL35 D1F12V1STK35 D1F12V1CST35 D1F12V1ORDI D1F12V1ORD D1F12V1NOR D1F12V1HIST
hdisk35	fcs0	
hdisk37	fcs1	D1F12V2ITEM D1F12V2WARE D1F12V2DIST D1F12V2NORI D1F12V2ORDI D1F12V2ORD D1F12V2NOR D1F12V2HIST
hdisk38	fcs0	
hdisk64	fcs14	D1F13V1ITEM D1F13V1WARE D1F13V1DIST D1F13V1NORI D1F13V1ORDI D1F13V1ORD D1F13V1NOR D1F13V1HIST
hdisk65	fcs13	
hdisk66	fcs14	D1F13V2ITEM D1F13V2WARE D1F13V2DIST D1F13V2NORI D1F13V2ORDI D1F13V2ORD D1F13V2NOR D1F13V2HIST
hdisk67	fcs13	
hdisk68	fcs14	D1F13V3ITEM D1F13V3WARE D1F13V3DIST D1F13V3NORI D1F13V3ORDI D1F13V3ORD D1F13V3NOR D1F13V3HIST
hdisk69	fcs13	
hdisk132	ssa3	db2SSAloglv
hdisk133		
hdisk134		
hdisk135		
hdisk136		
hdisk137		
hdisk73	ssa2	
hdisk127		
hdisk128		
hdisk129		
hdisk130		
hdisk131		

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	60,900	**tpmC**	763,898.39	**tpmC/W**	12.54
Table	**Rows**	**Data**	**Index**	**5% Space**	**Total Space**
History	2,035,800,000	125,268	0	0	125,268
Orders	2,035,800,000	74,677	56,848	0	131,525
Order-line	18,269,972,420	1,200,320	0	0	1,200,320
New-order	610,740,000	11,880	14,309	0	26,189
Warehouse	67,860	8	0	0	8
District	678,600	77	0	4	81
Item	100,000	11	0	1	11
Customer	1,827,000,000	1,427,510	88,200	75,786	1,591,496
Stock	6,090,000,000	1,982,575	0	99,129	2,081,704
Free Space	613,643				
Dynamic Space	1,412,145				
Static Space	3,673,299				
Daily Growth	283,412				
Daily Spread	188,526				
60 Days MB	31,989,554				
60 Days GB	31,240				
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	22,916,951.70	New Order transactions completed during the same 30 minute interval during stea			
Log per N-O txn	2.29	KB of log storage used per New Order transaction			
8 Hour Log (GB)	801.79	8-hour log storage required			
Disks Type	**Disk Formatted Capacity (GB)**	**SUT # of Disks**	**SUT Capacity (GB)**	**Priced # of Disks**	**Priced Capacity (GB)**
DB FAStT 18.2 GB	16.969	1,950	33,089	1,950	33,089
LOG SSA RAID5 (7x18.2GB)	101.813	12	1,222	12	1,222
OS SCSI 18.2 GB					33.94
Priced Space					34,345

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 %	1.4	1.14	1.18	1.12/2.26	1.14	0
Average	0.58	0.4	0.44	0.38/1.04	0.4	0
Maximum	20.39	17.41	14.44	9.60/9.25	10.6	15.01
			Think Times			
Minimum	0.01	0.01	0.01	0.01	0.01	N/A
Average	12.02	12.02	10.01	5.02	5.02	N/A
Maximum	120.20	120.20	100.10	50.20	50.20	N/A
			Keying Times			
Minimum	18.00	3.00	2.00	2.00	2.00	N/A
Average	18.01	3.01	2.01	2.01	2.01	N/A
Maximum	18.09	3.09	2.08	2.08	2.07	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

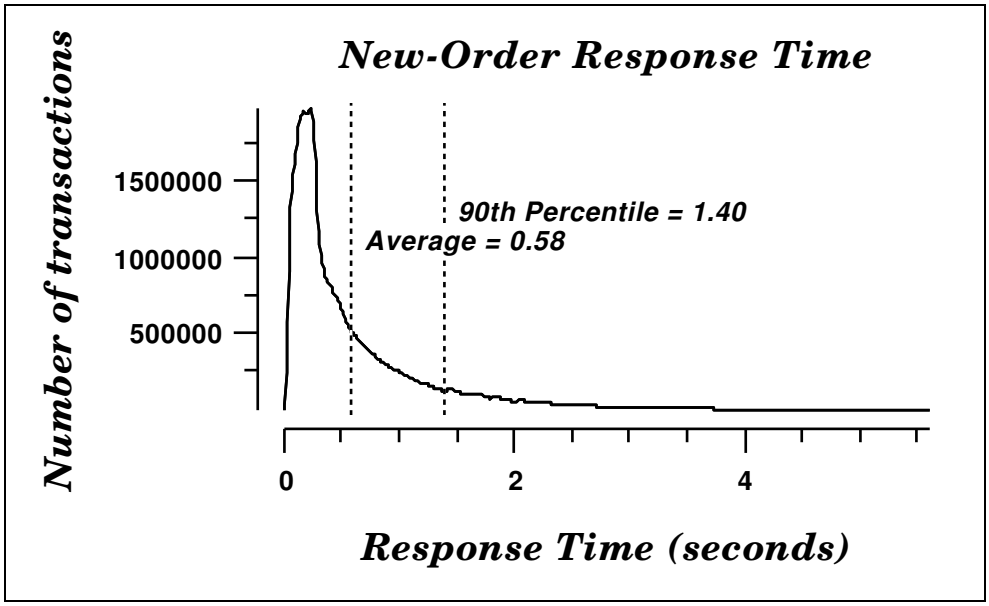


Figure 6-3-2. IBM eServer pseries 690 Turbo Model 7040-681 Payment Response Time Distribution

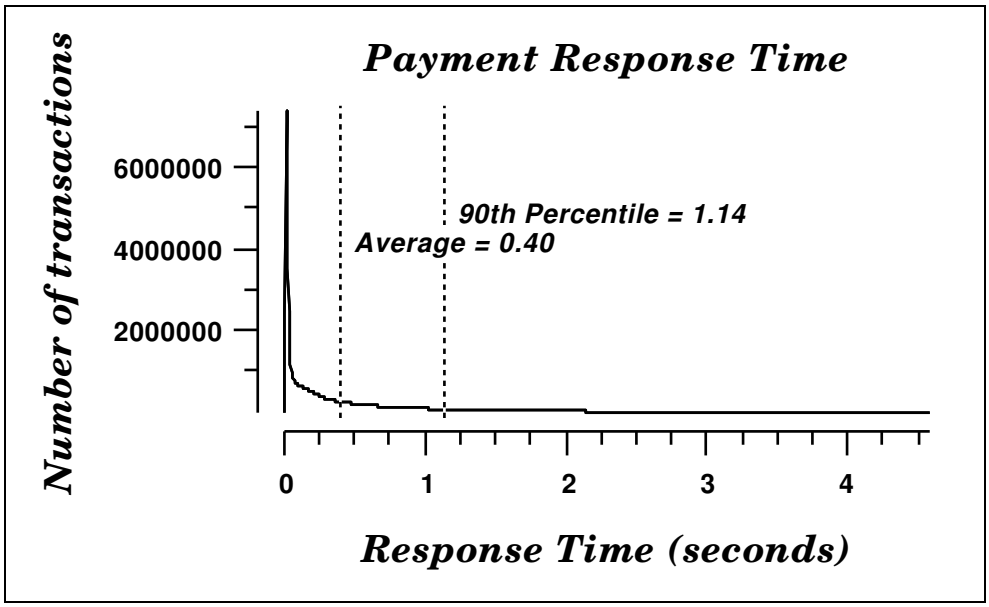


Figure 6-3-3. IBM eServer pseries 690 Turbo Model 7040-681 Order-Status Response Time Distribution

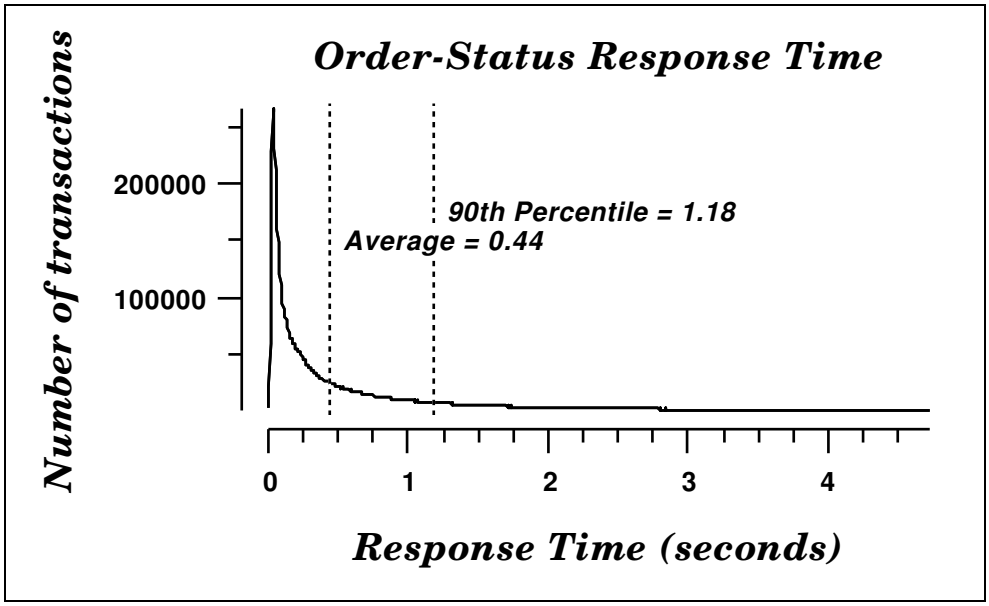


Figure 6-3-4. IBM eServer pSeries 690 Turbo Model 7040-681 Delivery (Interactive) Response Time Distribution

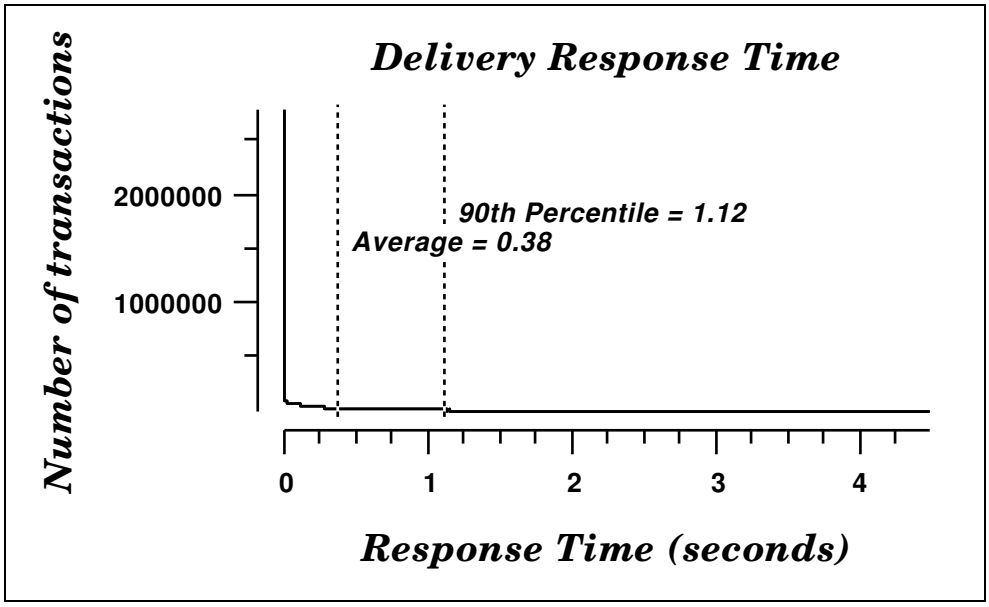


Figure 6-3-5. IBM eServer pSeries 690 Turbo Model 7040-681 Delivery (Deferred) Response Time Distribution

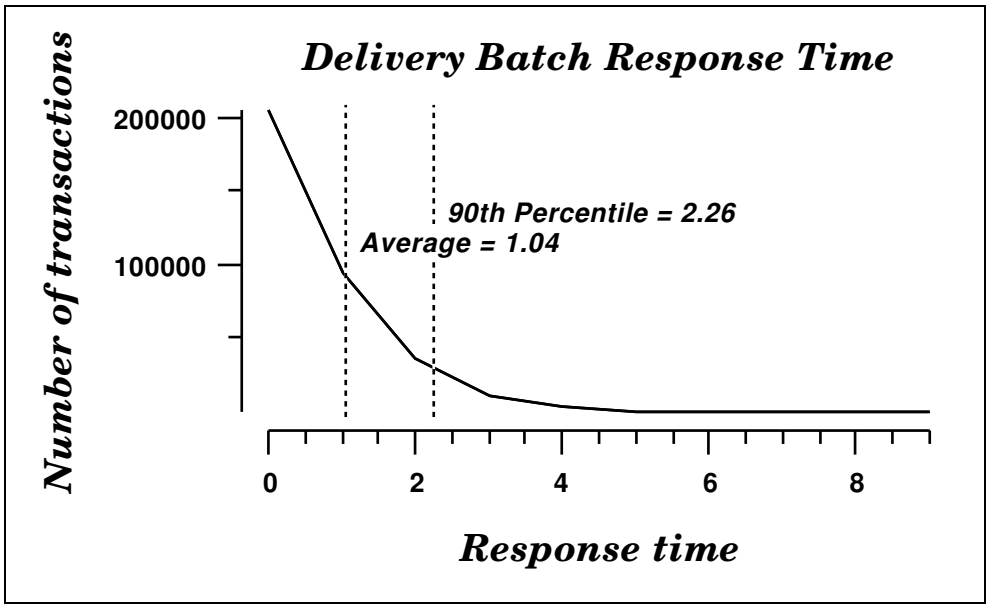
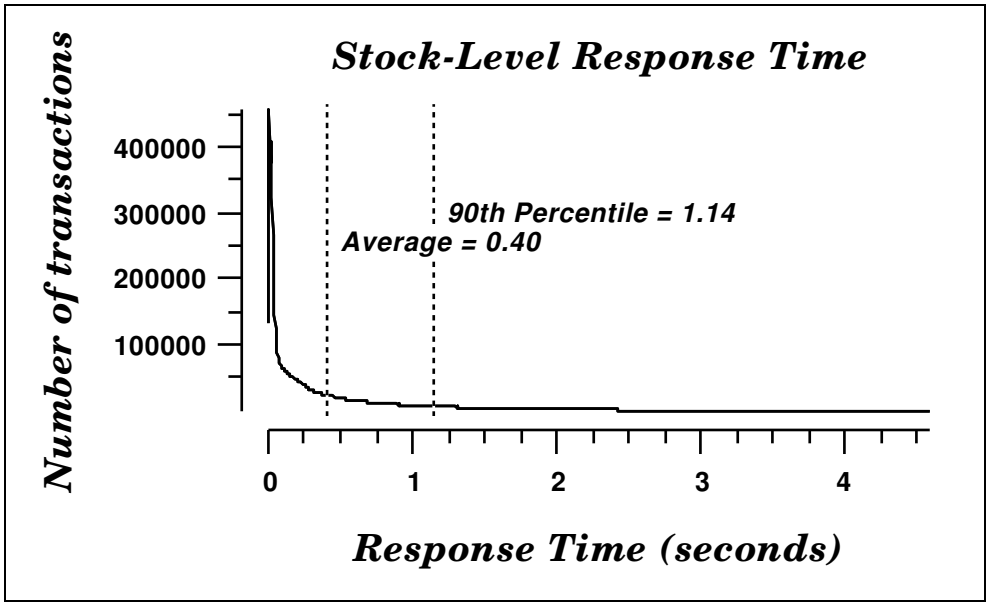


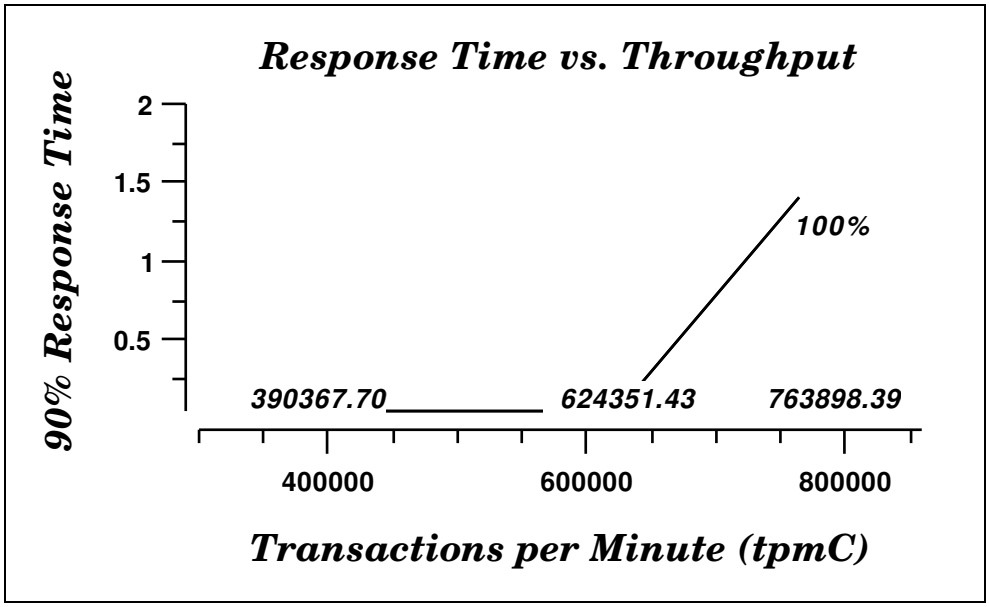
Figure 6-3-6. IBM eServer pSeries 690 Turbo Model 7040-681 Stock Level Response Time Distribution



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.

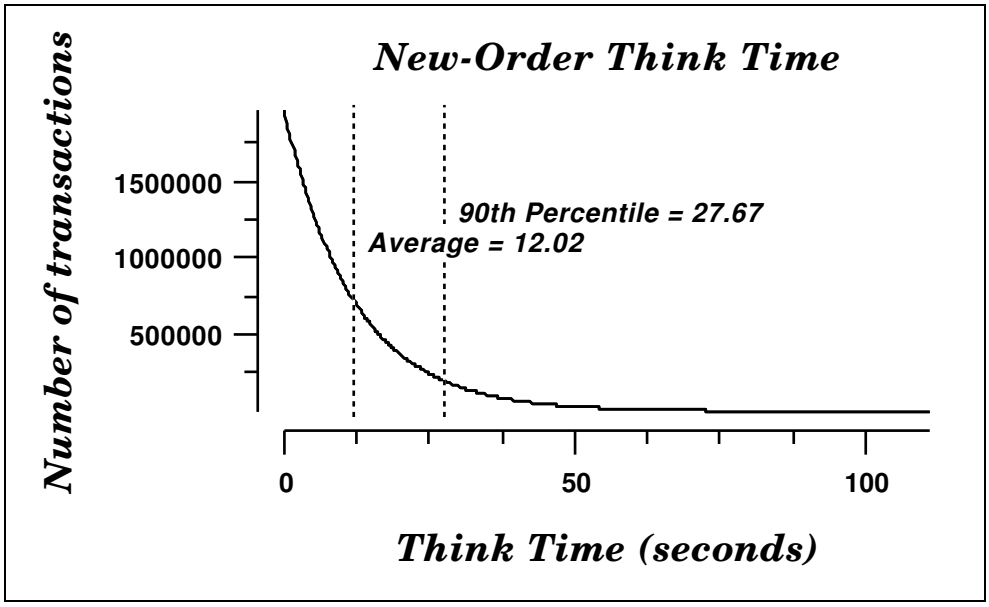
Figure 6-4-1. IBM eServer pSeries 690 Turbo Model 7040-681 New-Order Response Time vs. Throughput



6.5 Think Time Frequency Distribution

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

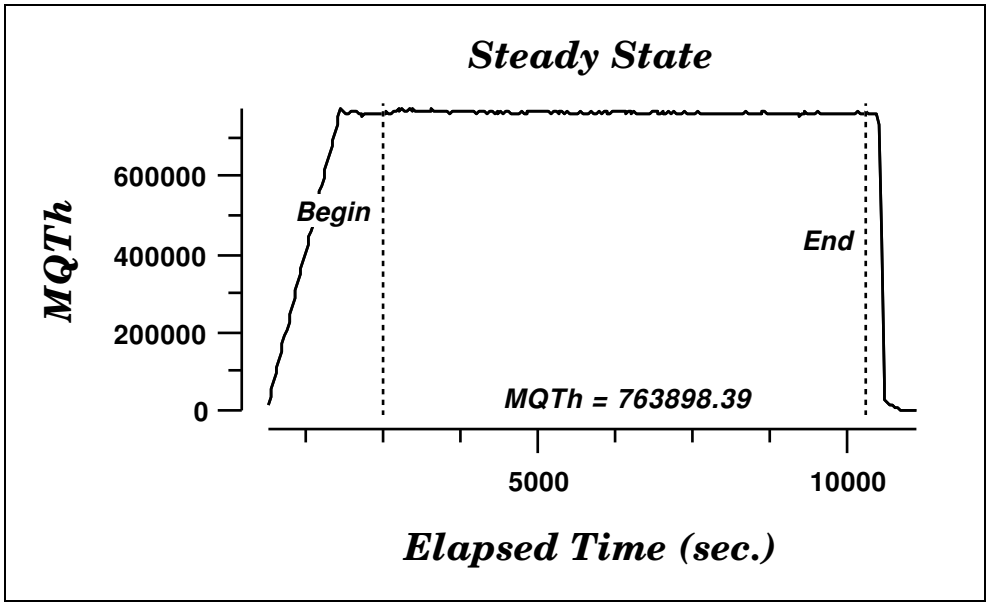
Figure 6-5-1. IBM eServer pSeries 690 Turbo Model 7040-681 New-Order Think Time Distribution



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 measurement interval of 2 hours 10 minutes was used to guarantee that all work normally performed during an 8-hour 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 type. Several server processes groups 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.

The measurement interval was 2 hours 10 minutes. No connections were lost during the measurement interval..

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 70 eServer pSeries 6F1 emulates a network of 609,000 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 725 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 725 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 43% 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
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	763,898.39	$6,301,741	$8.25	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

Sponsor: William A. Maron
IBM eServer Performance
11501 Burnet Road
Austin, TX 78758

Berni Schiefer
IBM DB2 Performance
8200 Warden Avenue
Markham, Ontario L6G-1C7

June 30, 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 DB2 8.1**
Transaction Manager: **BEA Tuxedo 8.0**

The results were:

CPU's Speed	Memory	Disks	NewOrder Response Time - 90%	tpmC
Server: IBM eServer pSeries 690 Turbo				
32 x POWER4+ (1.7 GHz)	512 GB (128 MB L3 cache/cpu)	3 x 18.2GB SCSI 96 x 18.2GB SSA 1950 x 18.2GB FAStT	1.40 Sec.	763,898.39
Thirty (30) Clients: IBM eServers pSeries 630-6E4 (Specification for each)				
4 x POWER4 (1.0 GHz)	16 GB (32 MB L3 cache/cpu)	1 x 18.2 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC requirements for Revision 5.1 of the benchmark.

The following verification items were given special attention:

- The transactions were correctly implemented.
- The database records were the proper size.
- The database was properly scaled and populated.
- The ACID properties were met.
- Input data was generated according to the specified percentages.
- The transaction cycle times included the required keying and think times.
- The reported response times were correctly measured.
- At least 90% of all delivery transactions met the 80 Second completion time limit.
- All 90% response times were under the specified maximums.
- The measurement interval was representative of steady state conditions.
- The reported measurement interval was 2 hours 10 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:

The measured system was configured using 36.4 GB spindles for 1446 of the FAStT disks drives. These were all substituted by 18.2 GB spindles in the priced configuration. Based on the specifications of these two types of spindles, the disk capacity used during the measurement and the results of I/O statistics collected, it is my opinion that this substitution had no material effect on the reported performance.

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

#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
**
*****************************************************************
*/

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
*****************************************************************
*/

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
**
**
**
**
** 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
**
**
** Returns	:
**		None
** Comments	:
**
*****************************************************************
*/

ShortField::ShortField(Terminal *ioP,int size,short *val) : Field(ioP,size),value(val)
{
```

int - xpos
int - ypos
int - size
char * - name

int - size
char * - name

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

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
****************************************************************
*/

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:
**
** Parameters :
**
** Returns		:
**
** Comments		:
**
****************************************************************
*/

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:
**
** Parameters :
**
** Returns		:
**
** 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:
**
** Parameters :
**
** Returns		:
**
** 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:
**
** Parameters :
**
** Returns		:
**
** Comments		:
**
****************************************************************
*/

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

		tpfree((char *) buffer);
}

/*
****************************************************************
** Name				: tuxClientConnect
** Description:
**
** Parameters :
**
** Returns		:
**
** Comments		:
**
****************************************************************
*/

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

		return OK;
}

/*
****************************************************************
** Name				: tuxClientDisconnect
** Description:
**
** Parameters :
**
** Returns		:
**
** 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)
{
        int thread_id = get_thread_id();

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

                dataP->header.thread_id=thread_id;

                int rc = tpcall("INFO",(char *) dataP,bufferSize,(char **)
&dataP,&bufferSize,0);
                *trpcStatus = rc;

                if (rc != OK)
                {
                        fprintf(stderr,"db_info tpcall err, rc: %d, tperrno: %d, %s\n",
                                                rc,tperrno,tpstrerror(tperrno));
                        dataP->header.sql_code          = RPC_ERROR;

                        return dataP;
                }

                //check the header to make sure we have valid data for this thread id
                if(dataP->header.thread_id != thread_id )
        {
                  fprintf(stderr,"db_info invalid buffer
error,header.sql_code:%d\n",dataP->header.sql_code);
                        dataP->header.sql_code          =
INVALID_BUFFER;

                        return dataP;
                }

                //dump message if new order was a fatal sql error/forced rollback
                if (dataP->header.sql_code != OK)
                {
                        fprintf(stderr,"db_info sql error,
header.sql_code:%d\n",dataP->header.sql_code);
                }
        }
        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)
{
        int thread_id   =       get_thread_id();
        int rc                          =       -1;

        if (null_db)
        {
                tux_new_order_null_db(dataP);
                *trpcStatus = 0; return dataP;
        }
        else
        {
                dataP->header.thread_id         =       thread_id;
                dataP->out.s_transtatus         =       INVALID_STATUS;

                rc = tpcall("ORDER",(char *) dataP,sizeof(newOrder_data_t),
                                                                        (char **)
&thread_context->tran_buffer,&thread_context->tran_buffer_size,0);
                *trpcStatus = rc;

                if (rc != OK)
                {
                        fprintf(stderr,"neworder tpcall err, w %d d %d cid %d oid %d
header.sql_code %d , rc: %d, tperrno: %d, %s\n",

dataP->in.s_W_ID,dataP->in.s_D_ID,dataP->in.s_C_ID,dataP->out.s_O_ID,dataP->header.sql_code,rc,tper
rno,tpstrerror(tperrno));

                        dataP->header.sql_code          = RPC_ERROR;
                        dataP->out.s_O_ID               = 9999;

                        return dataP;
                }

                //memcpy the result into the main allocated buffer

memcpy(dataP,thread_context->tran_buffer,thread_context->tran_buffer_size);

                //dump message if new order was a fatal sql error/forced rollback
                if (dataP->header.sql_code != OK)
                {
                        if (dataP->header.sql_code != INVALID_ITEM)
                        {
                                fprintf(stderr,"neworder sql error,w:%d d:%d
cid:%d oid %d header.sql_code %d\n",

 dataP->in.s_W_ID,dataP->in.s_D_ID,dataP->in.s_C_ID,dataP->out.s_O_ID,dataP->header.sql_code);
                                dataP->out.s_O_ID               =
9999;
                        }
                }

                //check the header to make sure we have valid data for this thread id
                if(dataP->header.thread_id != thread_id )
        {
                        fprintf(stderr,"neworder invalid buffer error, w %d d %d cid
%d oid %d header.sql_code %d ( thread_id %d header_thread_id %d
)\n",dataP->in.s_W_ID,dataP->in.s_D_ID,dataP->in.s_C_ID,dataP->out.s_O_ID,dataP->header.sql_code,
thread_id ,dataP->header.thread_id);
                        dataP->header.sql_code          =
INVALID_BUFFER;
                        dataP->out.s_O_ID               =
9999;
                }
        }
  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)
{

        int thread_id = get_thread_id();

        if (null_db)
        {
                tux_payment_null_db(dataP);
                *trpcStatus = 0; return dataP;
        }
        else
        {
                if (dataP->in.s_C_ID < 0)
```

```
    {
        fprintf(stderr,"Payment Error: invalid c_id from client, w_id:%d d_id%d c_id:%d amount:%lld
\n",dataP->in.s_W_ID,

dataP->in.s_D_ID,dataP->in.s_C_ID,dataP->in.s_H_AMOUNT);
        fflush(stderr);
    }

                                dataP->header.thread_id=thread_id;
                                dataP->out.s_transtatus=INVALID_STATUS;

                                int rc = tpcall("PAYMENT",(char *) dataP,sizeof(payment_data_t),

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

                                if(rc != OK)
                                {
                                                fprintf(stderr,"payment tpcall err,(w:%d d:%d
header.sql_code:%d) rc: %d, tperrno: %d, %s\n",

dataP->in.s_W_ID,dataP->in.s_D_ID,dataP->header.sql_code,rc,tperrno,tpstrerror(tperrno));

                                                dataP->header.sql_code          = RPC_ERROR;

                                                return dataP;
                                }

                                //memcpy the result into the main allocated buffer

memcpy(dataP,thread_context->tran_buffer,thread_context->tran_buffer_size);


                                //dump message if new order was a fatal sql error/forced rollback
    if (dataP->header.sql_code != OK)
    {
        fprintf(stderr,"payment sql error,w:%d d:%d header.sql_code:%d\n",
                    dataP->in.s_W_ID,dataP->in.s_D_ID,dataP->header.sql_code);
    }

                                //check the header to make sure we have valid data for this thread id
                                if(dataP->header.thread_id != thread_id )
                {
                                  fprintf(stderr,"payment invalid buffer error,w:%d d:%d
header.sql_code:%d\n",

dataP->in.s_W_ID,dataP->in.s_D_ID,dataP->header.sql_code);
                                                dataP->header.sql_code          =
INVALID_BUFFER;
                                }
                }
                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)
{
                int thread_id = get_thread_id();

                if (null_db)
                {
                                tux_order_status_null_db(dataP);
                                *trpcStatus = 0; return dataP;
                }
                else
                {
                                dataP->header.thread_id         =               thread_id;
                                dataP->out.s_transtatus         =               INVALID_STATUS;

                                int rc = tpcall("STATUS",(char *) dataP,sizeof(orderStatus_data_t),

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

                                if(rc != OK)
                                {
                                                fprintf(stderr,"order status tpcall err,(w:%d d:%d
header.sql_code:%d) rc: %d, tperrno: %d, %s\n",

dataP->in.s_W_ID,dataP->in.s_D_ID,dataP->header.sql_code,rc,tperrno,tpstrerror(tperrno));

                                                dataP->header.sql_code          = RPC_ERROR;

                                                return dataP;
                                }

                                //memcpy the result into the main allocated buffer

memcpy(dataP,thread_context->tran_buffer,thread_context->tran_buffer_size);

                                //dump message if new order was a fatal sql error/forced rollback
    if (dataP->header.sql_code != OK)
    {
        fprintf(stderr,"order status sql error,w:%d d:%d header.sql_code:%d\n",
                    dataP->in.s_W_ID,dataP->in.s_D_ID,dataP->header.sql_code);
    }

                                //check the header to make sure we have valid data for this thread id
                                if(dataP->header.thread_id != thread_id )
                {
                                  fprintf(stderr,"order status invalid buffer error,w:%d d:%d
header.sql_code:%d\n",

dataP->in.s_W_ID,dataP->in.s_D_ID,dataP->header.sql_code);
                                                dataP->header.sql_code          =
INVALID_BUFFER;

                                                return dataP;
                                }
                }
                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;

                int thread_id = get_thread_id();

                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;

                                dataP->header.thread_id         =               thread_id;
                                dataP->header.sql_code          =               OK;
                                dataP->out.s_transtatus         =  OK;

                                int rc = tpacall("DELIVERY",(char *)
dataP,sizeof(delivery_data_t),TPNOREPLY);
                                *trpcStatus = rc;

                                if(rc != OK)
                                {
                                                fprintf(stderr,"delivery tuxedo tpcall err,(w:%d
header.sql_code:%d) rc: %d, tperrno: %d, %s\n",

dataP->in.s_W_ID,dataP->header.sql_code,rc,tperrno,tpstrerror(tperrno));

                                                dataP->header.sql_code          = RPC_ERROR;

                                                return dataP;
                                }

                                //dump message if new order was a fatal sql error/forced rollback
    if (dataP->header.sql_code != OK)
    {
        fprintf(stderr,"delivery sql error,w:%d header.sql_code:%d\n",
                    dataP->in.s_W_ID,dataP->header.sql_code);
    }

                                //check the header to make sure we have valid data for this thread id
```

```
                        if(dataP->header.thread_id != thread_id )
                {
                        fprintf(stderr,"delivery invalid buffer error,w:%d header.sql_code:%d\n",

dataP->in.s_W_ID,dataP->header.sql_code);

                                dataP->header.sql_code          =
INVALID_BUFFER;
                        }
                }
                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)
{
                int thread_id = get_thread_id();

                if (null_db)
                {
                        tux_stock_level_null_db(dataP);
                        *trpcStatus = 0; return dataP;
                }
                else
                {
   dataP->out.s_transtatus=INVALID_STATUS;
                        dataP->header.thread_id=thread_id;

                        int rc = tpcall("STOCK",(char *) dataP,sizeof(stockLevel_data_t),

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

                        if(rc != OK)
                        {
                                fprintf(stderr,"stock tuxedo tpcall err,(w:%d d:%d
header.sql_code:%d) rc: %d, tperrno: %d, %s\n",

dataP->in.s_W_ID,dataP->in.s_D_ID,dataP->header.sql_code,rc,tperrno,tpstrerror(tperrno));

                                dataP->header.sql_code          = RPC_ERROR;

                                return dataP;
                        }

                        //memcpy the result into the main allocated buffer

memcpy(dataP,thread_context->tran_buffer,thread_context->tran_buffer_size);

                        //dump message if new order was a fatal sql error/forced rollback
   if (dataP->header.sql_code != OK)
   {
      fprintf(stderr,"stock sql error,w:%d d:%d header.sql_code:%d\n",
                 dataP->in.s_W_ID,dataP->in.s_D_ID,dataP->header.sql_code);
   }

                        //check the header to make sure we have valid data for this thread id
                        if(dataP->header.thread_id != thread_id )
                {
                        fprintf(stderr,"stock invalid buffer error,w:%d d:%d header.sql_code:%d\n",

dataP->in.s_W_ID,dataP->in.s_D_ID,dataP->header.sql_code);

                                dataP->header.sql_code          =
INVALID_BUFFER;
                        }
                }
                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
****************************************************************
*/

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
**
***************************************************************
*/

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**
*****************************************************************
*/

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;
		case INVALID_STATUS:
				threadP->write("ERROR: Rollback -- INVALID_STATUS");
				break;
		case INVALID_BUFFER:
				threadP->write("ERROR: Rollback -- INVALID_BUFFER");
				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_NEW_RC
				if (data->out.s_transtatus != OK)
				{
						fprintf(stderr,"New Order Error, in NewOrder::respond()
oid_buffer:%s , s_transtatus %d w_id %d d_id %d c_id %d o_id %d\n",

buf,data->out.s_transtatus,data->in.s_W_ID,data->in.s_D_ID,data->in.s_C_ID,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,15,6);		format_int(buf,10,data->out.s_O_ID);

threadP->write(buf,9);
		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:
**
** Parameters :
**
** Returns		:
**
** 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:
**
** Parameters :
**
**
** Returns		:
**
** Comments		:
**
*************************************************************
*/

Payment::Payment(User_data *udP,Thread_data *threadP) : Screen(udP,threadP)
{
		data
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;

//		fields[2]->set_blank_with_prev();

// Name only without customer #
		return;
}

/*
*************************************************************
** Name				: Payment::validate
** Description:
**
** Parameters :
**
** Returns		:
**
** 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;

//		fields[2]->set_blank_with_prev();

		// Name only without customer #
		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 :
**
** Parameters :
**
** Returns		:
**
** Comments		:
**
****************************************************************
*/

inline TerminalBuffer::TerminalBuffer(int size)
{
		BufSize		= size;
		buffer		= new char [BufSize];

		Pos		= 0;
		Start		= 0;
		num_marks	= 0;

		return;
}

/*
****************************************************************
** Name		: TerminalBuffer::mark
** Description :
**
** Parameters :
**
** Returns		:
**
** 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 :
**
** Parameters :
**
** Returns		:
**
** 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 :
**
** Parameters :
**
** Returns		:
**
** 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
*****************************************************************
*/

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 :
**
**
** Parameters :
**
**
** Returns	:
**
**
** 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 :
**
**
** Parameters :
**
**
** Returns	:
**
**
** 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\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)
				{
						if(pData->out.s_transtatus != INVALID_ITEM)
						{
								fprintf(dump_log,"New Order SQL Error,
s_transtatus %d w_id %d d_id %d c_id %d o_id %d\n",pData->out.s_transtatus,

pData->in.s_W_ID,pData->in.s_D_ID,pData->in.s_C_ID,pData->out.s_O_ID);
								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,"Payment Error: invalid payment amount
from client, w_id:%d d_id%d c_id:%d
amount:%lld\n",pData->in.s_W_ID,pData->in.s_D_ID,pData->in.s_C_ID,pData->in.s_H_AMOUNT);

						fflush(dump_log);
				}
				else if (pData->in.s_C_ID < 0)
		{
						fprintf(dump_log,"Payment Error: invalid c_id from client,
w_id:%d d_id%d c_id:%d amount:%lld \n",pData->in.s_W_ID,pData->in.s_D_ID,pData->in.s_C_ID,pData
->in.s_H_AMOUNT);
						fflush(dump_log);
				}
		#endif

		if(!null_db)
		{
				payment_sql(&pData->in,&pData->out);
		}
		else tux_payment_null_db(pData);

		#ifdef 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: w_id:%d d_id:%d
c_id:%d s_tran:%d \n",pData->in.s_W_ID,pData->in.s_D_ID,pData->in.s_C_ID,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: w_id:%d carrier:%d
s_tran:%d \n",pData->in.s_W_ID,pData->in.s_O_CARRIER_ID,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                                    : 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");
}


/*
*****************************************************************
** 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)
                {
                                //hex_dump_stock(pData);
                                stocklev_sql(&pData->in,&pData->out);
                                //hex_dump_stock(pData);
                }
                else tux_stock_level_null_db(pData);

                #if DUMP_DATA
                                if (pData->out.s_transtatus != OK)
                                {
                                                fprintf(dump_log,"Stock Error: w_id:%d d_id:%d
s_tran:%d\n",pData->in.s_W_ID,pData->in.s_D_ID,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");

}

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);

}



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

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 <tpccdbg.h>
#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                                   1

//////////////////////////////////////////////////////////////////
// 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 2003
** Classification : IBM Internal Use Only
**
** History       :
**              Developed at IBM Austin by the AIX RS/6000
**              performance group
**
** 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'

#define INVALID_STATUS    0xABCDEF
#define INVALID_BUFFER    0xFEDCBA

//////////////////////////////////////////////////////////////////
// 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          1
```

```
#define DEBUG_LOG                       1
//#define ERROR_LOG         1

#define DUMP_CLIENT_DATA                0
#define CHECK_NEW_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;

                INT32b                                          thread_id;

                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;

//////////////////////////////////////////////////////////////////////
// Inline functions
//////////////////////////////////////////////////////////////////////

inline int get_thread_id()
{
                return (int) pthread_self();
}

#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              319
#define C_OL_I_ID           3849
#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

```
#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 60900
#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
```

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,"\t\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
```

tpcc_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  )
                             , (  SMALLINT( 7 )        , :id6  , :ol_quantity6  , :supply_w_id6  )

                             ) 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( 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 )

                             ) 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) = 10240
Utilities heap size (4KB)              (UTIL_HEAP_SZ) = 20000
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) = 65000
Default application heap (4KB)           (APPLHEAPSZ) = 2500
Package cache size (4KB)                (PCKCACHESZ) = 50000
Statistics heap size (4KB)             (STAT_HEAP_SZ) = 100000
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) = 1600
Average number of active applications       (AVG_APPLS) = 1
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/rdbSSAloglv
Overflow log path                     (OVERFLOWLOGPATH) =
Mirror log path                       (MIRRORLOGPATH) =
First active log file                                 = S0000063.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) = 0
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) = 1600
Agent pool size                       (NUM_POOLAGENTS) = 1600
Initial number of agents in pool       (NUM_INITAGENTS) = 1600
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=19 MAX=28 CLOPT="-s
ORDER:tux_new_order" RQADDR=newo
tuxServer SRVGRP=GROUP2 SRVID=2001 MIN=19 MAX=24 CLOPT="-s
PAYMENT:tux_payment" RQADDR=pmt
tuxServer SRVGRP=GROUP3 SRVID=3001 MIN=2 MAX=8 CLOPT="-s
STATUS:tux_order_status" RQADDR=ostat
tuxServer SRVGRP=GROUP4 SRVID=4001 MIN=10 MAX=13 CLOPT="-s
DELIVERY:tux_delivery -- -d" RQADDR=dlvry
tuxServer SRVGRP=GROUP5 SRVID=5001 MIN=2 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

```
SW_dist_intr false          Enable SW distribution of interrupts          True
autorestart  true           Automatically REBOOT system after a crash     True
boottype     disk           N/A                                           False
conslogin    enable         System Console Login                          False
cpuguard     enable         CPU Guard                                     True
frequency    568000000      System Bus Frequency                          False
fullcore     false          Enable full CORE dump                         True
fwversion    IBM,RG030415_d54e03_s Firmware version and revision levels    False
iostat       false          Continuously maintain DISK I/O history        True
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
maxpout      0              HIGH water mark for pending write I/Os per file True
maxuproc     5000           Maximum number of PROCESSES allowed per user  True
minpout      0              LOW water mark for pending write I/Os per file True
modelname    IBM,7040-681   Machine name                                  False
ncargs       6              ARG/ENV list size in 4K byte blocks           True
pre430core   false          Use pre-430 style CORE dump                   True
pre520tune   disable        Pre-520 tuning compatibility mode             True
realmem      536870912      Amount of usable physical memory in Kbytes    False
rtasversion  1              Open Firmware RTAS version                    False
systemid     IBM,010ykrh0v  Hardware system identifier                    False
```

lsattr -El aio0 -a maxreqs
maxreqs 300000 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       3079K        3079K                4KB pages        S
--------------------------------------------------------------------------------
maxfree               128    128    128    16     200K   4KB pages        D
   minfree
   memory_frames
--------------------------------------------------------------------------------
minfree               120    120    120    8      200K   4KB pages        D
   maxfree
   memory_frames
--------------------------------------------------------------------------------
minperm%              1      20     1      1      100    % memory         D
   maxperm%
--------------------------------------------------------------------------------
minperm               39624        39624                                  S
--------------------------------------------------------------------------------
maxperm%              80     80     80     1      100    % memory         D
   minperm%
   maxclient%
--------------------------------------------------------------------------------
maxperm               3095K        3095K                                  S
--------------------------------------------------------------------------------
strict_maxperm        0      0      0      0      1      boolean          D
--------------------------------------------------------------------------------
maxpin%               98     98     98     1      99     % memory         D
   pinnable_frames
   memory_frames
--------------------------------------------------------------------------------
maxpin                128450K      128450K                                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             31856K       31856K               4KB pages        S
--------------------------------------------------------------------------------
npskill               254848 254848 254848 1      31855K 4KB pages        D
--------------------------------------------------------------------------------
npswarn               995K   995K   995K   0      31855K 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          31008  0      31008  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_customer_29 prefetchsize 0;
alter tablespace is_customer_30 prefetchsize 0;
alter tablespace is_customer_31 prefetchsize 0;
alter tablespace is_customer_32 prefetchsize 0;
alter tablespace is_customer_33 prefetchsize 0;
alter tablespace is_customer_34 prefetchsize 0;
alter tablespace is_customer_35 prefetchsize 0;
alter tablespace is_customer_36 prefetchsize 0;
alter tablespace is_customer_37 prefetchsize 0;
alter tablespace is_customer_38 prefetchsize 0;
alter tablespace is_customer_39 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_customer_29 prefetchsize 0;
alter tablespace ts_customer_30 prefetchsize 0;
alter tablespace ts_customer_31 prefetchsize 0;
alter tablespace ts_customer_32 prefetchsize 0;
alter tablespace ts_customer_33 prefetchsize 0;
alter tablespace ts_customer_34 prefetchsize 0;
alter tablespace ts_customer_35 prefetchsize 0;
alter tablespace ts_customer_36 prefetchsize 0;
alter tablespace ts_customer_37 prefetchsize 0;
alter tablespace ts_customer_38 prefetchsize 0;
alter tablespace ts_customer_39 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_orderline_29 prefetchsize 0;
alter tablespace ts_orderline_30 prefetchsize 0;
alter tablespace ts_orderline_31 prefetchsize 0;
alter tablespace ts_orderline_32 prefetchsize 0;
alter tablespace ts_orderline_33 prefetchsize 0;
alter tablespace ts_orderline_34 prefetchsize 0;
alter tablespace ts_orderline_35 prefetchsize 0;
alter tablespace ts_orderline_36 prefetchsize 0;
alter tablespace ts_orderline_37 prefetchsize 0;
alter tablespace ts_orderline_38 prefetchsize 0;
alter tablespace ts_orderline_39 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_stock_29 prefetchsize 0;
alter tablespace ts_stock_30 prefetchsize 0;
alter tablespace ts_stock_31 prefetchsize 0;
alter tablespace ts_stock_32 prefetchsize 0;
alter tablespace ts_stock_33 prefetchsize 0;
alter tablespace ts_stock_34 prefetchsize 0;
alter tablespace ts_stock_35 prefetchsize 0;
alter tablespace ts_stock_36 prefetchsize 0;
alter tablespace ts_stock_37 prefetchsize 0;
alter tablespace ts_stock_38 prefetchsize 0;
alter tablespace ts_stock_39 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_customer_29 prefetchsize 512;
alter tablespace is_customer_30 prefetchsize 512;
alter tablespace is_customer_31 prefetchsize 512;
alter tablespace is_customer_32 prefetchsize 512;
alter tablespace is_customer_33 prefetchsize 512;
alter tablespace is_customer_34 prefetchsize 512;
alter tablespace is_customer_35 prefetchsize 512;
alter tablespace is_customer_36 prefetchsize 512;
alter tablespace is_customer_37 prefetchsize 512;
alter tablespace is_customer_38 prefetchsize 512;
alter tablespace is_customer_39 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_customer_29 prefetchsize 512;
alter tablespace ts_customer_30 prefetchsize 512;
alter tablespace ts_customer_31 prefetchsize 512;
alter tablespace ts_customer_32 prefetchsize 512;
alter tablespace ts_customer_33 prefetchsize 512;
alter tablespace ts_customer_34 prefetchsize 512;
alter tablespace ts_customer_35 prefetchsize 512;
alter tablespace ts_customer_36 prefetchsize 512;
alter tablespace ts_customer_37 prefetchsize 512;
alter tablespace ts_customer_38 prefetchsize 512;
alter tablespace ts_customer_39 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_orderline_29 prefetchsize 512;
alter tablespace ts_orderline_30 prefetchsize 512;
alter tablespace ts_orderline_31 prefetchsize 512;
alter tablespace ts_orderline_32 prefetchsize 512;
alter tablespace ts_orderline_33 prefetchsize 512;
alter tablespace ts_orderline_34 prefetchsize 512;
alter tablespace ts_orderline_35 prefetchsize 512;
alter tablespace ts_orderline_36 prefetchsize 512;
alter tablespace ts_orderline_37 prefetchsize 512;
alter tablespace ts_orderline_38 prefetchsize 512;
alter tablespace ts_orderline_39 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_stock_29 prefetchsize 512;
alter tablespace ts_stock_30 prefetchsize 512;
alter tablespace ts_stock_31 prefetchsize 512;
alter tablespace ts_stock_32 prefetchsize 512;
alter tablespace ts_stock_33 prefetchsize 512;
alter tablespace ts_stock_34 prefetchsize 512;
alter tablespace ts_stock_35 prefetchsize 512;
alter tablespace ts_stock_36 prefetchsize 512;
alter tablespace ts_stock_37 prefetchsize 512;
alter tablespace ts_stock_38 prefetchsize 512;
alter tablespace ts_stock_39 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 1740);
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
1741 AND 3480);
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
3481 AND 5220);
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
5221 AND 6960);
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
6961 AND 8700);
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
8701 AND 10440);
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
10441 AND 12180);
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
12181 AND 13920);
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
13921 AND 15660);
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 15661 AND 17400);
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 17401 AND 19140);
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 19141 AND 20880);
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 20881 AND 22620);
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 22621 AND 24360);
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 24361 AND 26100);
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 26101 AND 27840);
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 27841 AND 29580);
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 29581 AND 31320);
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 31321 AND 33060);
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 33061 AND 34800);
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 34801 AND 36540);
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 36541 AND 38280);
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 38281 AND 40020);
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 40021 AND 41760);
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 41761 AND 43500);
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 43501 AND 45240);
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 45241 AND 46980);
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
BETWEEN 46981 AND 48720);
SET INTEGRITY FOR CUSTOMER28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR CUSTOMER29 OFF;
ALTER TABLE CUSTOMER29 DROP CONSTRAINT CUSTOMER29CKC;
ALTER TABLE CUSTOMER29 ADD CONSTRAINT CUSTOMER29CKC CHECK (C_W_ID
BETWEEN 48721 AND 50460);
SET INTEGRITY FOR CUSTOMER29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR CUSTOMER30 OFF;
ALTER TABLE CUSTOMER30 DROP CONSTRAINT CUSTOMER30CKC;
ALTER TABLE CUSTOMER30 ADD CONSTRAINT CUSTOMER30CKC CHECK (C_W_ID
BETWEEN 50461 AND 52200);
SET INTEGRITY FOR CUSTOMER30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR CUSTOMER31 OFF;
ALTER TABLE CUSTOMER31 DROP CONSTRAINT CUSTOMER31CKC;
ALTER TABLE CUSTOMER31 ADD CONSTRAINT CUSTOMER31CKC CHECK (C_W_ID
BETWEEN 52201 AND 53940);
SET INTEGRITY FOR CUSTOMER31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR CUSTOMER32 OFF;
ALTER TABLE CUSTOMER32 DROP CONSTRAINT CUSTOMER32CKC;
ALTER TABLE CUSTOMER32 ADD CONSTRAINT CUSTOMER32CKC CHECK (C_W_ID
BETWEEN 53941 AND 55680);
```

```
SET INTEGRITY FOR CUSTOMER32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR CUSTOMER33 OFF;
ALTER TABLE CUSTOMER33 DROP CONSTRAINT CUSTOMER33CKC;
ALTER TABLE CUSTOMER33 ADD CONSTRAINT CUSTOMER33CKC CHECK (C_W_ID
BETWEEN 55681 AND 57420);
SET INTEGRITY FOR CUSTOMER33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR CUSTOMER34 OFF;
ALTER TABLE CUSTOMER34 DROP CONSTRAINT CUSTOMER34CKC;
ALTER TABLE CUSTOMER34 ADD CONSTRAINT CUSTOMER34CKC CHECK (C_W_ID
BETWEEN 57421 AND 59160);
SET INTEGRITY FOR CUSTOMER34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR CUSTOMER35 OFF;
ALTER TABLE CUSTOMER35 DROP CONSTRAINT CUSTOMER35CKC;
ALTER TABLE CUSTOMER35 ADD CONSTRAINT CUSTOMER35CKC CHECK (C_W_ID >=
59161);
SET INTEGRITY FOR CUSTOMER35 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 1740);
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 1741 AND 3480);
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 3481 AND 5220);
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 5221 AND 6960);
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 6961 AND 8700);
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 8701 AND 10440);
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 10441 AND 12180);
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 12181 AND 13920);
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 13921 AND 15660);
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 15661 AND 17400);
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 17401 AND 19140);
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 19141 AND 20880);
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 20881 AND 22620);
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 22621 AND 24360);
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 24361 AND 26100);
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 26101 AND 27840);
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 27841 AND 29580);
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 29581 AND 31320);
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 31321 AND 33060);
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 33061 AND 34800);
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 34801 AND 36540);
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 36541 AND 38280);
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 38281 AND 40020);
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 40021 AND 41760);
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 41761 AND 43500);
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 43501 AND 45240);
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 45241 AND 46980);
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
BETWEEN 46981 AND 48720);
SET INTEGRITY FOR ORDER_LINE28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR ORDER_LINE29 OFF;
ALTER TABLE ORDER_LINE29 DROP CONSTRAINT ORDER_LINE29CKC;
ALTER TABLE ORDER_LINE29 ADD CONSTRAINT ORDER_LINE29CKC CHECK (OL_W_ID
BETWEEN 48721 AND 50460);
SET INTEGRITY FOR ORDER_LINE29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR ORDER_LINE30 OFF;
ALTER TABLE ORDER_LINE30 DROP CONSTRAINT ORDER_LINE30CKC;
ALTER TABLE ORDER_LINE30 ADD CONSTRAINT ORDER_LINE30CKC CHECK (OL_W_ID
BETWEEN 50461 AND 52200);
SET INTEGRITY FOR ORDER_LINE30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR ORDER_LINE31 OFF;
ALTER TABLE ORDER_LINE31 DROP CONSTRAINT ORDER_LINE31CKC;
ALTER TABLE ORDER_LINE31 ADD CONSTRAINT ORDER_LINE31CKC CHECK (OL_W_ID
BETWEEN 52201 AND 53940);
SET INTEGRITY FOR ORDER_LINE31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR ORDER_LINE32 OFF;
ALTER TABLE ORDER_LINE32 DROP CONSTRAINT ORDER_LINE32CKC;
ALTER TABLE ORDER_LINE32 ADD CONSTRAINT ORDER_LINE32CKC CHECK (OL_W_ID
BETWEEN 53941 AND 55680);
SET INTEGRITY FOR ORDER_LINE32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR ORDER_LINE33 OFF;
ALTER TABLE ORDER_LINE33 DROP CONSTRAINT ORDER_LINE33CKC;
ALTER TABLE ORDER_LINE33 ADD CONSTRAINT ORDER_LINE33CKC CHECK (OL_W_ID
BETWEEN 55681 AND 57420);
SET INTEGRITY FOR ORDER_LINE33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR ORDER_LINE34 OFF;
ALTER TABLE ORDER_LINE34 DROP CONSTRAINT ORDER_LINE34CKC;
ALTER TABLE ORDER_LINE34 ADD CONSTRAINT ORDER_LINE34CKC CHECK (OL_W_ID
BETWEEN 57421 AND 59160);
SET INTEGRITY FOR ORDER_LINE34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR ORDER_LINE35 OFF;
ALTER TABLE ORDER_LINE35 DROP CONSTRAINT ORDER_LINE35CKC;
ALTER TABLE ORDER_LINE35 ADD CONSTRAINT ORDER_LINE35CKC CHECK (OL_W_ID >=
59161);
 SET INTEGRITY FOR ORDER_LINE35 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
1740);
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 1741 AND
3480);
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 3481 AND
5220);
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 5221 AND
6960);
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 6961 AND
8700);
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 8701 AND
10440);
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 10441
AND 12180);
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 12181
AND 13920);
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 13921
AND 15660);
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 15661
AND 17400);
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 17401
AND 19140);
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 19141
AND 20880);
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 20881
AND 22620);
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 22621
AND 24360);
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 24361
AND 26100);
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 26101
AND 27840);
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 27841
AND 29580);
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 29581
AND 31320);
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 31321
AND 33060);
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 33061
AND 34800);
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 34801
AND 36540);
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 36541
AND 38280);
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 38281
AND 40020);
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 40021
AND 41760);
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 41761
AND 43500);
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 43501
AND 45240);
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 45241
AND 46980);
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 BETWEEN 46981
AND 48720);
SET INTEGRITY FOR STOCK28 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR STOCK29 OFF;
ALTER TABLE STOCK29 DROP CONSTRAINT STOCK29CKC;
ALTER TABLE STOCK29 ADD CONSTRAINT STOCK29CKC CHECK (S_W_ID BETWEEN 48721
AND 50460);
SET INTEGRITY FOR STOCK29 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR STOCK30 OFF;
ALTER TABLE STOCK30 DROP CONSTRAINT STOCK30CKC;
ALTER TABLE STOCK30 ADD CONSTRAINT STOCK30CKC CHECK (S_W_ID BETWEEN 50461
AND 52200);
SET INTEGRITY FOR STOCK30 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR STOCK31 OFF;
ALTER TABLE STOCK31 DROP CONSTRAINT STOCK31CKC;
ALTER TABLE STOCK31 ADD CONSTRAINT STOCK31CKC CHECK (S_W_ID BETWEEN 52201
AND 53940);
SET INTEGRITY FOR STOCK31 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR STOCK32 OFF;
ALTER TABLE STOCK32 DROP CONSTRAINT STOCK32CKC;
ALTER TABLE STOCK32 ADD CONSTRAINT STOCK32CKC CHECK (S_W_ID BETWEEN 53941
AND 55680);
SET INTEGRITY FOR STOCK32 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR STOCK33 OFF;
ALTER TABLE STOCK33 DROP CONSTRAINT STOCK33CKC;
ALTER TABLE STOCK33 ADD CONSTRAINT STOCK33CKC CHECK (S_W_ID BETWEEN 55681
AND 57420);
SET INTEGRITY FOR STOCK33 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR STOCK34 OFF;
ALTER TABLE STOCK34 DROP CONSTRAINT STOCK34CKC;
ALTER TABLE STOCK34 ADD CONSTRAINT STOCK34CKC CHECK (S_W_ID BETWEEN 57421
AND 59160);
SET INTEGRITY FOR STOCK34 ALL IMMEDIATE UNCHECKED;
connect reset;
connect to tpcc in share mode;
SET INTEGRITY FOR STOCK35 OFF;
ALTER TABLE STOCK35 DROP CONSTRAINT STOCK35CKC;
ALTER TABLE STOCK35 ADD CONSTRAINT STOCK35CKC CHECK (S_W_ID >= 59161);
SET INTEGRITY FOR STOCK35 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;
connect to tpcc in share mode;
DROP INDEX CUST_IDXB29;
CREATE INDEX CUST_IDXB29
        ON CUSTOMER29(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_IDXB30;
CREATE INDEX CUST_IDXB30
        ON CUSTOMER30(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_IDXB31;
CREATE INDEX CUST_IDXB31
        ON CUSTOMER31(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_IDXB32;
CREATE INDEX CUST_IDXB32
        ON CUSTOMER32(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_IDXB33;
CREATE INDEX CUST_IDXB33
        ON CUSTOMER33(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_IDXB34;
CREATE INDEX CUST_IDXB34
        ON CUSTOMER34(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_IDXB35;
CREATE INDEX CUST_IDXB35
        ON CUSTOMER35(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_IDXB36;
CREATE INDEX CUST_IDXB36
        ON CUSTOMER36(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_IDXB37;
CREATE INDEX CUST_IDXB37
        ON CUSTOMER37(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_IDXB38;
CREATE INDEX CUST_IDXB38
        ON CUSTOMER38(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_IDXB39;
CREATE INDEX CUST_IDXB39
        ON CUSTOMER39(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 1740,
        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 1741 ENDING AT 3480,
        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 3481 ENDING AT 5220,
        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 5221 ENDING AT 6960,
        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 6961 ENDING AT 8700,
        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 8701 ENDING AT 10440,
        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 10441 ENDING AT 12180,
        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 12181 ENDING AT 13920,
        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 13921 ENDING AT 15660,
        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 15661 ENDING AT 17400,
        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 17401 ENDING AT 19140,
        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 19141 ENDING AT 20880,
        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 20881 ENDING AT 22620,
         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 22621 ENDING AT 24360,
         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 24361 ENDING AT 26100,
         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 26101 ENDING AT 27840,
         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 27841 ENDING AT 29580,
         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 29581 ENDING AT 31320,
        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 31321 ENDING AT 33060,
        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 33061 ENDING AT 34800,
        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 34801 ENDING AT 36540,
        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 36541 ENDING AT 38280,
        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 38281 ENDING AT 40020,
        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 40021 ENDING AT 41760,
        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 41761 ENDING AT 43500,
        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 43501 ENDING AT 45240,
        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 45241 ENDING AT 46980,
        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 46981 ENDING AT 48720,
        C_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE CUSTOMER29;
CREATE TABLE CUSTOMER29
        (
        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_29
        INDEX IN is_customer_29
        ORGANIZE BY KEY SEQUENCE (
        C_ID STARTING FROM 1 ENDING AT 3000,
        C_W_ID STARTING FROM 48721 ENDING AT 50460,
```

```
            C_D_ID STARTING FROM 1 ENDING AT 10
           )
           ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE CUSTOMER30;
CREATE TABLE CUSTOMER30
           (
            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_30
           INDEX IN is_customer_30
           ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 50461 ENDING AT 52200,
            C_D_ID STARTING FROM 1 ENDING AT 10
           )
           ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE CUSTOMER31;
CREATE TABLE CUSTOMER31
           (
            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_31
           INDEX IN is_customer_31
           ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 52201 ENDING AT 53940,
            C_D_ID STARTING FROM 1 ENDING AT 10
           )
           ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE CUSTOMER32;
CREATE TABLE CUSTOMER32
           (
            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_32
           INDEX IN is_customer_32
           ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 53941 ENDING AT 55680,
            C_D_ID STARTING FROM 1 ENDING AT 10
           )
           ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE CUSTOMER33;
CREATE TABLE CUSTOMER33
           (
            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_33
           INDEX IN is_customer_33
           ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 55681 ENDING AT 57420,
            C_D_ID STARTING FROM 1 ENDING AT 10
           )
           ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE CUSTOMER34;
CREATE TABLE CUSTOMER34
           (
            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_34
           INDEX IN is_customer_34
           ORGANIZE BY KEY SEQUENCE (
            C_ID STARTING FROM 1 ENDING AT 3000,
            C_W_ID STARTING FROM 57421 ENDING AT 59160,
            C_D_ID STARTING FROM 1 ENDING AT 10
           )
           ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE CUSTOMER35;
CREATE TABLE CUSTOMER35
           (
            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_35
        INDEX IN is_customer_35
        ORGANIZE BY KEY SEQUENCE (
        C_ID STARTING FROM 1 ENDING AT 3000,
        C_W_ID STARTING FROM 59161 ENDING AT 60900,
        C_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE CUSTOMER36;
CREATE TABLE CUSTOMER36
        (
        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_36
        INDEX IN is_customer_36
        ORGANIZE BY KEY SEQUENCE (
        C_ID STARTING FROM 1 ENDING AT 3000,
        C_W_ID STARTING FROM 60901 ENDING AT 62640,
        C_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE CUSTOMER37;
CREATE TABLE CUSTOMER37
        (
        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_37
        INDEX IN is_customer_37
        ORGANIZE BY KEY SEQUENCE (
        C_ID STARTING FROM 1 ENDING AT 3000,
        C_W_ID STARTING FROM 62641 ENDING AT 64380,
        C_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE CUSTOMER38;
CREATE TABLE CUSTOMER38
        (
        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_38
        INDEX IN is_customer_38
        ORGANIZE BY KEY SEQUENCE (
        C_ID STARTING FROM 1 ENDING AT 3000,
        C_W_ID STARTING FROM 64381 ENDING AT 66120,
        C_D_ID STARTING FROM 1 ENDING AT 10
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE CUSTOMER39;
CREATE TABLE CUSTOMER39
        (
        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_39
        INDEX IN is_customer_39
        ORGANIZE BY KEY SEQUENCE (
        C_ID STARTING FROM 1 ENDING AT 3000,
        C_W_ID STARTING FROM 66121 ENDING AT 67860,
        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 67860
        )
        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 3772,
         O_W_ID STARTING FROM 1 ENDING AT 67860,
         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 1740,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3772,
         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 1741 ENDING AT 3480,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3772,
         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 3481 ENDING AT 5220,
         OL_D_ID STARTING FROM 1 ENDING AT 10,
         OL_O_ID STARTING FROM 1 ENDING AT 3772,
         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 5221 ENDING AT 6960,
 OL_D_ID STARTING FROM 1 ENDING AT 10,
 OL_O_ID STARTING FROM 1 ENDING AT 3772,
 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 6961 ENDING AT 8700,
 OL_D_ID STARTING FROM 1 ENDING AT 10,
 OL_O_ID STARTING FROM 1 ENDING AT 3772,
 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 8701 ENDING AT 10440,
 OL_D_ID STARTING FROM 1 ENDING AT 10,
 OL_O_ID STARTING FROM 1 ENDING AT 3772,
 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 10441 ENDING AT 12180,
 OL_D_ID STARTING FROM 1 ENDING AT 10,
 OL_O_ID STARTING FROM 1 ENDING AT 3772,
 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 12181 ENDING AT 13920,
 OL_D_ID STARTING FROM 1 ENDING AT 10,
 OL_O_ID STARTING FROM 1 ENDING AT 3772,
 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 13921 ENDING AT 15660,
 OL_D_ID STARTING FROM 1 ENDING AT 10,
 OL_O_ID STARTING FROM 1 ENDING AT 3772,
 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 15661 ENDING AT 17400,
 OL_D_ID STARTING FROM 1 ENDING AT 10,
 OL_O_ID STARTING FROM 1 ENDING AT 3772,
 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 17401 ENDING AT 19140,
 OL_D_ID STARTING FROM 1 ENDING AT 10,
 OL_O_ID STARTING FROM 1 ENDING AT 3772,
 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 19141 ENDING AT 20880,
        OL_D_ID STARTING FROM 1 ENDING AT 10,
        OL_O_ID STARTING FROM 1 ENDING AT 3772,
        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 20881 ENDING AT 22620,
        OL_D_ID STARTING FROM 1 ENDING AT 10,
        OL_O_ID STARTING FROM 1 ENDING AT 3772,
        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 22621 ENDING AT 24360,
        OL_D_ID STARTING FROM 1 ENDING AT 10,
        OL_O_ID STARTING FROM 1 ENDING AT 3772,
        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 24361 ENDING AT 26100,
        OL_D_ID STARTING FROM 1 ENDING AT 10,
        OL_O_ID STARTING FROM 1 ENDING AT 3772,
        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 26101 ENDING AT 27840,
        OL_D_ID STARTING FROM 1 ENDING AT 10,
        OL_O_ID STARTING FROM 1 ENDING AT 3772,
        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 27841 ENDING AT 29580,
        OL_D_ID STARTING FROM 1 ENDING AT 10,
        OL_O_ID STARTING FROM 1 ENDING AT 3772,
        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 29581 ENDING AT 31320,
        OL_D_ID STARTING FROM 1 ENDING AT 10,
        OL_O_ID STARTING FROM 1 ENDING AT 3772,
        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 31321 ENDING AT 33060,
        OL_D_ID STARTING FROM 1 ENDING AT 10,
        OL_O_ID STARTING FROM 1 ENDING AT 3772,
        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 33061 ENDING AT 34800,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            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 34801 ENDING AT 36540,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            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 36541 ENDING AT 38280,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            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 38281 ENDING AT 40020,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            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 40021 ENDING AT 41760,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            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 41761 ENDING AT 43500,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            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 43501 ENDING AT 45240,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            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 45241 ENDING AT 46980,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            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 46981 ENDING AT 48720,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE ORDER_LINE29;
CREATE TABLE ORDER_LINE29
            (
            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_29
            INDEX IN ts_orderline_29
            ORGANIZE BY KEY SEQUENCE (
            OL_W_ID STARTING FROM 48721 ENDING AT 50460,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE ORDER_LINE30;
CREATE TABLE ORDER_LINE30
            (
            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_30
            INDEX IN ts_orderline_30
            ORGANIZE BY KEY SEQUENCE (
            OL_W_ID STARTING FROM 50461 ENDING AT 52200,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE ORDER_LINE31;
CREATE TABLE ORDER_LINE31
            (
            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_31
            INDEX IN ts_orderline_31
            ORGANIZE BY KEY SEQUENCE (
            OL_W_ID STARTING FROM 52201 ENDING AT 53940,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE ORDER_LINE32;
CREATE TABLE ORDER_LINE32
            (
            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_32
            INDEX IN ts_orderline_32
            ORGANIZE BY KEY SEQUENCE (
            OL_W_ID STARTING FROM 53941 ENDING AT 55680,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE ORDER_LINE33;
CREATE TABLE ORDER_LINE33
            (
            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_33
            INDEX IN ts_orderline_33
            ORGANIZE BY KEY SEQUENCE (
            OL_W_ID STARTING FROM 55681 ENDING AT 57420,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE ORDER_LINE34;
CREATE TABLE ORDER_LINE34
            (
            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_34
            INDEX IN ts_orderline_34
            ORGANIZE BY KEY SEQUENCE (
            OL_W_ID STARTING FROM 57421 ENDING AT 59160,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE ORDER_LINE35;
CREATE TABLE ORDER_LINE35
            (
            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_35
            INDEX IN ts_orderline_35
            ORGANIZE BY KEY SEQUENCE (
            OL_W_ID STARTING FROM 59161 ENDING AT 60900,
            OL_D_ID STARTING FROM 1 ENDING AT 10,
            OL_O_ID STARTING FROM 1 ENDING AT 3772,
            OL_NUMBER STARTING FROM 1 ENDING AT 15
            )
            ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE ORDER_LINE36;
CREATE TABLE ORDER_LINE36
```

```
(
 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_36
INDEX IN ts_orderline_36
ORGANIZE BY KEY SEQUENCE (
 OL_W_ID STARTING FROM 60901 ENDING AT 62640,
 OL_D_ID STARTING FROM 1 ENDING AT 10,
 OL_O_ID STARTING FROM 1 ENDING AT 3772,
 OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE ORDER_LINE37;
CREATE TABLE ORDER_LINE37
(
 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_37
INDEX IN ts_orderline_37
ORGANIZE BY KEY SEQUENCE (
 OL_W_ID STARTING FROM 62641 ENDING AT 64380,
 OL_D_ID STARTING FROM 1 ENDING AT 10,
 OL_O_ID STARTING FROM 1 ENDING AT 3772,
 OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE ORDER_LINE38;
CREATE TABLE ORDER_LINE38
(
 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_38
INDEX IN ts_orderline_38
ORGANIZE BY KEY SEQUENCE (
 OL_W_ID STARTING FROM 64381 ENDING AT 66120,
 OL_D_ID STARTING FROM 1 ENDING AT 10,
 OL_O_ID STARTING FROM 1 ENDING AT 3772,
 OL_NUMBER STARTING FROM 1 ENDING AT 15
)
ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE ORDER_LINE39;
CREATE TABLE ORDER_LINE39
(
 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_39
INDEX IN ts_orderline_39
ORGANIZE BY KEY SEQUENCE (
 OL_W_ID STARTING FROM 66121 ENDING AT 67860,
 OL_D_ID STARTING FROM 1 ENDING AT 10,
 OL_O_ID STARTING FROM 1 ENDING AT 3772,
 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 1740
)
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 1741 ENDING AT 3480
)
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 3481 ENDING AT 5220
)
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 5221 ENDING AT 6960
        )
        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 6961 ENDING AT 8700
        )
        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 8701 ENDING AT 10440
        )
        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 10441 ENDING AT 12180
        )
        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 12181 ENDING AT 13920
        )
        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 13921 ENDING AT 15660
        )
        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 15661 ENDING AT 17400
        )
        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 17401 ENDING AT 19140
        )
        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 19141 ENDING AT 20880
        )
        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 20881 ENDING AT 22620
        )
        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 22621 ENDING AT 24360
        )
        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 24361 ENDING AT 26100
        )
        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 26101 ENDING AT 27840
        )
        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 27841 ENDING AT 29580
        )
        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 29581 ENDING AT 31320
        )
        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 31321 ENDING AT 33060
        )
        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 33061 ENDING AT 34800
        )
        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 34801 ENDING AT 36540
        )
        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 36541 ENDING AT 38280
        )
        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 38281 ENDING AT 40020
            )
            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 40021 ENDING AT 41760
            )
            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 41761 ENDING AT 43500
            )
            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 43501 ENDING AT 45240
            )
            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 45241 ENDING AT 46980
            )
            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 46981 ENDING AT 48720
            )
            ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE STOCK29;
CREATE TABLE STOCK29
            (
            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_29
            INDEX IN ts_stock_29
            ORGANIZE BY KEY SEQUENCE (
            S_I_ID STARTING FROM 1 ENDING AT 100000,
            S_W_ID STARTING FROM 48721 ENDING AT 50460
            )
            ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE STOCK30;
CREATE TABLE STOCK30
            (
            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_30
        INDEX IN ts_stock_30
        ORGANIZE BY KEY SEQUENCE (
        S_I_ID STARTING FROM 1 ENDING AT 100000,
        S_W_ID STARTING FROM 50461 ENDING AT 52200
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE STOCK31;
CREATE TABLE STOCK31
        (
        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_31
        INDEX IN ts_stock_31
        ORGANIZE BY KEY SEQUENCE (
        S_I_ID STARTING FROM 1 ENDING AT 100000,
        S_W_ID STARTING FROM 52201 ENDING AT 53940
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE STOCK32;
CREATE TABLE STOCK32
        (
        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_32
        INDEX IN ts_stock_32
        ORGANIZE BY KEY SEQUENCE (
        S_I_ID STARTING FROM 1 ENDING AT 100000,
        S_W_ID STARTING FROM 53941 ENDING AT 55680
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE STOCK33;
CREATE TABLE STOCK33
        (
        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_33
        INDEX IN ts_stock_33
        ORGANIZE BY KEY SEQUENCE (
        S_I_ID STARTING FROM 1 ENDING AT 100000,
        S_W_ID STARTING FROM 55681 ENDING AT 57420
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE STOCK34;
CREATE TABLE STOCK34
        (
        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_34
        INDEX IN ts_stock_34
        ORGANIZE BY KEY SEQUENCE (
        S_I_ID STARTING FROM 1 ENDING AT 100000,
        S_W_ID STARTING FROM 57421 ENDING AT 59160
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE STOCK35;
CREATE TABLE STOCK35
        (
        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_35
        INDEX IN ts_stock_35
        ORGANIZE BY KEY SEQUENCE (
        S_I_ID STARTING FROM 1 ENDING AT 100000,
        S_W_ID STARTING FROM 59161 ENDING AT 60900
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE STOCK36;
CREATE TABLE STOCK36
        (
        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_36
        INDEX IN ts_stock_36
        ORGANIZE BY KEY SEQUENCE (
        S_I_ID STARTING FROM 1 ENDING AT 100000,
        S_W_ID STARTING FROM 60901 ENDING AT 62640
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
```

```
DROP TABLE STOCK37;
CREATE TABLE STOCK37
        (
         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_37
        INDEX IN ts_stock_37
        ORGANIZE BY KEY SEQUENCE (
         S_I_ID STARTING FROM 1 ENDING AT 100000,
         S_W_ID STARTING FROM 62641 ENDING AT 64380
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE STOCK38;
CREATE TABLE STOCK38
        (
         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_38
        INDEX IN ts_stock_38
        ORGANIZE BY KEY SEQUENCE (
         S_I_ID STARTING FROM 1 ENDING AT 100000,
         S_W_ID STARTING FROM 64381 ENDING AT 66120
        )
        ALLOW OVERFLOW;
connect reset;
connect to tpcc in share mode;
DROP TABLE STOCK39;
CREATE TABLE STOCK39
        (
         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_39
        INDEX IN ts_stock_39
        ORGANIZE BY KEY SEQUENCE (
         S_I_ID STARTING FROM 1 ENDING AT 100000,
         S_W_ID STARTING FROM 66121 ENDING AT 67860
        )
        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 67860
        )
        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 1740) UNION ALL SELECT * FROM CUSTOMER2
        WHERE (C_W_ID BETWEEN 1741 AND 3480) UNION ALL SELECT * FROM
CUSTOMER3
        WHERE (C_W_ID BETWEEN 3481 AND 5220) UNION ALL SELECT * FROM
CUSTOMER4
        WHERE (C_W_ID BETWEEN 5221 AND 6960) UNION ALL SELECT * FROM
CUSTOMER5
        WHERE (C_W_ID BETWEEN 6961 AND 8700) UNION ALL SELECT * FROM
CUSTOMER6
        WHERE (C_W_ID BETWEEN 8701 AND 10440) UNION ALL SELECT * FROM
CUSTOMER7
        WHERE (C_W_ID BETWEEN 10441 AND 12180) UNION ALL SELECT * FROM
CUSTOMER8
        WHERE (C_W_ID BETWEEN 12181 AND 13920) UNION ALL SELECT * FROM
CUSTOMER9
        WHERE (C_W_ID BETWEEN 13921 AND 15660) UNION ALL SELECT * FROM
CUSTOMER10
        WHERE (C_W_ID BETWEEN 15661 AND 17400) UNION ALL SELECT * FROM
CUSTOMER11
        WHERE (C_W_ID BETWEEN 17401 AND 19140) UNION ALL SELECT * FROM
CUSTOMER12
        WHERE (C_W_ID BETWEEN 19141 AND 20880) UNION ALL SELECT * FROM
CUSTOMER13
        WHERE (C_W_ID BETWEEN 20881 AND 22620) UNION ALL SELECT * FROM
CUSTOMER14
        WHERE (C_W_ID BETWEEN 22621 AND 24360) UNION ALL SELECT * FROM
CUSTOMER15
        WHERE (C_W_ID BETWEEN 24361 AND 26100) UNION ALL SELECT * FROM
CUSTOMER16
        WHERE (C_W_ID BETWEEN 26101 AND 27840) UNION ALL SELECT * FROM
CUSTOMER17
        WHERE (C_W_ID BETWEEN 27841 AND 29580) UNION ALL SELECT * FROM
CUSTOMER18
        WHERE (C_W_ID BETWEEN 29581 AND 31320) UNION ALL SELECT * FROM
CUSTOMER19
        WHERE (C_W_ID BETWEEN 31321 AND 33060) UNION ALL SELECT * FROM
CUSTOMER20
        WHERE (C_W_ID BETWEEN 33061 AND 34800) UNION ALL SELECT * FROM
CUSTOMER21
        WHERE (C_W_ID BETWEEN 34801 AND 36540) UNION ALL SELECT * FROM
CUSTOMER22
        WHERE (C_W_ID BETWEEN 36541 AND 38280) UNION ALL SELECT * FROM
CUSTOMER23
        WHERE (C_W_ID BETWEEN 38281 AND 40020) UNION ALL SELECT * FROM
CUSTOMER24
        WHERE (C_W_ID BETWEEN 40021 AND 41760) UNION ALL SELECT * FROM
CUSTOMER25
        WHERE (C_W_ID BETWEEN 41761 AND 43500) UNION ALL SELECT * FROM
```

```
CUSTOMER26
            WHERE (C_W_ID BETWEEN 43501 AND 45240) UNION ALL SELECT * FROM
CUSTOMER27
            WHERE (C_W_ID BETWEEN 45241 AND 46980) UNION ALL SELECT * FROM
CUSTOMER28
            WHERE (C_W_ID BETWEEN 46981 AND 48720) UNION ALL SELECT * FROM
CUSTOMER29
            WHERE (C_W_ID BETWEEN 48721 AND 50460) UNION ALL SELECT * FROM
CUSTOMER30
            WHERE (C_W_ID BETWEEN 50461 AND 52200) UNION ALL SELECT * FROM
CUSTOMER31
            WHERE (C_W_ID BETWEEN 52201 AND 53940) UNION ALL SELECT * FROM
CUSTOMER32
            WHERE (C_W_ID BETWEEN 53941 AND 55680) UNION ALL SELECT * FROM
CUSTOMER33
            WHERE (C_W_ID BETWEEN 55681 AND 57420) UNION ALL SELECT * FROM
CUSTOMER34
            WHERE (C_W_ID BETWEEN 57421 AND 59160) UNION ALL SELECT * FROM
CUSTOMER35
            WHERE (C_W_ID >= 59161) 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 1740) UNION ALL SELECT * FROM
ORDER_LINE2
             WHERE (OL_W_ID BETWEEN 1741 AND 3480) UNION ALL SELECT * FROM
ORDER_LINE3
             WHERE (OL_W_ID BETWEEN 3481 AND 5220) UNION ALL SELECT * FROM
ORDER_LINE4
             WHERE (OL_W_ID BETWEEN 5221 AND 6960) UNION ALL SELECT * FROM
ORDER_LINE5
             WHERE (OL_W_ID BETWEEN 6961 AND 8700) UNION ALL SELECT * FROM
ORDER_LINE6
             WHERE (OL_W_ID BETWEEN 8701 AND 10440) UNION ALL SELECT * FROM
ORDER_LINE7
             WHERE (OL_W_ID BETWEEN 10441 AND 12180) UNION ALL SELECT * FROM
ORDER_LINE8
             WHERE (OL_W_ID BETWEEN 12181 AND 13920) UNION ALL SELECT * FROM
ORDER_LINE9
             WHERE (OL_W_ID BETWEEN 13921 AND 15660) UNION ALL SELECT * FROM
ORDER_LINE10
             WHERE (OL_W_ID BETWEEN 15661 AND 17400) UNION ALL SELECT * FROM
ORDER_LINE11
             WHERE (OL_W_ID BETWEEN 17401 AND 19140) UNION ALL SELECT * FROM
ORDER_LINE12
             WHERE (OL_W_ID BETWEEN 19141 AND 20880) UNION ALL SELECT * FROM
ORDER_LINE13
             WHERE (OL_W_ID BETWEEN 20881 AND 22620) UNION ALL SELECT * FROM
ORDER_LINE14
             WHERE (OL_W_ID BETWEEN 22621 AND 24360) UNION ALL SELECT * FROM
ORDER_LINE15
             WHERE (OL_W_ID BETWEEN 24361 AND 26100) UNION ALL SELECT * FROM
ORDER_LINE16
             WHERE (OL_W_ID BETWEEN 26101 AND 27840) UNION ALL SELECT * FROM
ORDER_LINE17
             WHERE (OL_W_ID BETWEEN 27841 AND 29580) UNION ALL SELECT * FROM
ORDER_LINE18
             WHERE (OL_W_ID BETWEEN 29581 AND 31320) UNION ALL SELECT * FROM
ORDER_LINE19
             WHERE (OL_W_ID BETWEEN 31321 AND 33060) UNION ALL SELECT * FROM
ORDER_LINE20
             WHERE (OL_W_ID BETWEEN 33061 AND 34800) UNION ALL SELECT * FROM
ORDER_LINE21
             WHERE (OL_W_ID BETWEEN 34801 AND 36540) UNION ALL SELECT * FROM
ORDER_LINE22
             WHERE (OL_W_ID BETWEEN 36541 AND 38280) UNION ALL SELECT * FROM
ORDER_LINE23
             WHERE (OL_W_ID BETWEEN 38281 AND 40020) UNION ALL SELECT * FROM
ORDER_LINE24
             WHERE (OL_W_ID BETWEEN 40021 AND 41760) UNION ALL SELECT * FROM
ORDER_LINE25
             WHERE (OL_W_ID BETWEEN 41761 AND 43500) UNION ALL SELECT * FROM
ORDER_LINE26
             WHERE (OL_W_ID BETWEEN 43501 AND 45240) UNION ALL SELECT * FROM
ORDER_LINE27
             WHERE (OL_W_ID BETWEEN 45241 AND 46980) UNION ALL SELECT * FROM
ORDER_LINE28
             WHERE (OL_W_ID BETWEEN 46981 AND 48720) UNION ALL SELECT * FROM
ORDER_LINE29
             WHERE (OL_W_ID BETWEEN 48721 AND 50460) UNION ALL SELECT * FROM
ORDER_LINE30
             WHERE (OL_W_ID BETWEEN 50461 AND 52200) UNION ALL SELECT * FROM
ORDER_LINE31
             WHERE (OL_W_ID BETWEEN 52201 AND 53940) UNION ALL SELECT * FROM
ORDER_LINE32
             WHERE (OL_W_ID BETWEEN 53941 AND 55680) UNION ALL SELECT * FROM
ORDER_LINE33
             WHERE (OL_W_ID BETWEEN 55681 AND 57420) UNION ALL SELECT * FROM
ORDER_LINE34
             WHERE (OL_W_ID BETWEEN 57421 AND 59160) UNION ALL SELECT * FROM
ORDER_LINE35
             WHERE (OL_W_ID >= 59161) 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 1740) UNION ALL SELECT * FROM STOCK2
            WHERE (S_W_ID BETWEEN 1741 AND 3480) UNION ALL SELECT * FROM STOCK3
            WHERE (S_W_ID BETWEEN 3481 AND 5220) UNION ALL SELECT * FROM STOCK4
            WHERE (S_W_ID BETWEEN 5221 AND 6960) UNION ALL SELECT * FROM STOCK5
            WHERE (S_W_ID BETWEEN 6961 AND 8700) UNION ALL SELECT * FROM STOCK6
            WHERE (S_W_ID BETWEEN 8701 AND 10440) UNION ALL SELECT * FROM STOCK7
            WHERE (S_W_ID BETWEEN 10441 AND 12180) UNION ALL SELECT * FROM STOCK8
            WHERE (S_W_ID BETWEEN 12181 AND 13920) UNION ALL SELECT * FROM STOCK9
            WHERE (S_W_ID BETWEEN 13921 AND 15660) UNION ALL SELECT * FROM
STOCK10
            WHERE (S_W_ID BETWEEN 15661 AND 17400) UNION ALL SELECT * FROM
STOCK11
            WHERE (S_W_ID BETWEEN 17401 AND 19140) UNION ALL SELECT * FROM
STOCK12
            WHERE (S_W_ID BETWEEN 19141 AND 20880) UNION ALL SELECT * FROM
STOCK13
            WHERE (S_W_ID BETWEEN 20881 AND 22620) UNION ALL SELECT * FROM
STOCK14
            WHERE (S_W_ID BETWEEN 22621 AND 24360) UNION ALL SELECT * FROM
STOCK15
            WHERE (S_W_ID BETWEEN 24361 AND 26100) UNION ALL SELECT * FROM
STOCK16
            WHERE (S_W_ID BETWEEN 26101 AND 27840) UNION ALL SELECT * FROM
STOCK17
            WHERE (S_W_ID BETWEEN 27841 AND 29580) UNION ALL SELECT * FROM
STOCK18
            WHERE (S_W_ID BETWEEN 29581 AND 31320) UNION ALL SELECT * FROM
STOCK19
            WHERE (S_W_ID BETWEEN 31321 AND 33060) UNION ALL SELECT * FROM
STOCK20
            WHERE (S_W_ID BETWEEN 33061 AND 34800) UNION ALL SELECT * FROM
STOCK21
            WHERE (S_W_ID BETWEEN 34801 AND 36540) UNION ALL SELECT * FROM
STOCK22
            WHERE (S_W_ID BETWEEN 36541 AND 38280) UNION ALL SELECT * FROM
STOCK23
            WHERE (S_W_ID BETWEEN 38281 AND 40020) UNION ALL SELECT * FROM
STOCK24
            WHERE (S_W_ID BETWEEN 40021 AND 41760) UNION ALL SELECT * FROM
STOCK25
            WHERE (S_W_ID BETWEEN 41761 AND 43500) UNION ALL SELECT * FROM
STOCK26
            WHERE (S_W_ID BETWEEN 43501 AND 45240) UNION ALL SELECT * FROM
STOCK27
            WHERE (S_W_ID BETWEEN 45241 AND 46980) UNION ALL SELECT * FROM
STOCK28
            WHERE (S_W_ID BETWEEN 46981 AND 48720) UNION ALL SELECT * FROM
STOCK29
            WHERE (S_W_ID BETWEEN 48721 AND 50460) UNION ALL SELECT * FROM
STOCK30
            WHERE (S_W_ID BETWEEN 50461 AND 52200) UNION ALL SELECT * FROM
```

```
STOCK31
        WHERE (S_W_ID BETWEEN 52201 AND 53940) UNION ALL SELECT * FROM
STOCK32
        WHERE (S_W_ID BETWEEN 53941 AND 55680) UNION ALL SELECT * FROM
STOCK33
        WHERE (S_W_ID BETWEEN 55681 AND 57420) UNION ALL SELECT * FROM
STOCK34
        WHERE (S_W_ID BETWEEN 57421 AND 59160) UNION ALL SELECT * FROM
STOCK35
        WHERE (S_W_ID >= 59161) WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

GEN_CUSTOMER.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1 435 -f1 /flats/ua_01/customer_01_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 436 870 -f1 /flats/ua_01/customer_01_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 871 1305 -f1 /flats/ua_01/customer_01_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1306 1740 -f1 /flats/ua_01/customer_01_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 1741 2175 -f1 /flats/ua_02/customer_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2176 2610 -f1 /flats/ua_02/customer_02_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 2611 3045 -f1 /flats/ua_02/customer_02_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3046 3480 -f1 /flats/ua_02/customer_02_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3481 3915 -f1 /flats/ua_03/customer_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 3916 4350 -f1 /flats/ua_03/customer_03_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4351 4785 -f1 /flats/ua_03/customer_03_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 4786 5220 -f1 /flats/ua_03/customer_03_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5221 5655 -f1 /flats/ua_04/customer_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 5656 6090 -f1 /flats/ua_04/customer_04_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6091 6525 -f1 /flats/ua_04/customer_04_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6526 6960 -f1 /flats/ua_04/customer_04_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 6961 7395 -f1 /flats/ua_05/customer_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7396 7830 -f1 /flats/ua_05/customer_05_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 7831 8265 -f1 /flats/ua_05/customer_05_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8266 8700 -f1 /flats/ua_05/customer_05_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 8701 9135 -f1 /flats/ua_06/customer_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9136 9570 -f1 /flats/ua_06/customer_06_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 9571 10005 -f1 /flats/ua_06/customer_06_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10006 10440 -f1 /flats/ua_06/customer_06_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10441 10875 -f1 /flats/ua_07/customer_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 10876 11310 -f1 /flats/ua_07/customer_07_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 11311 11745 -f1 /flats/ua_07/customer_07_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 11746 12180 -f1 /flats/ua_07/customer_07_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12181 12615 -f1 /flats/ua_08/customer_08_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 12616 13050 -f1 /flats/ua_08/customer_08_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 13051 13485 -f1 /flats/ua_08/customer_08_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 13486 13920 -f1 /flats/ua_08/customer_08_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 13921 14355 -f1 /flats/ua_09/customer_09_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 14356 14790 -f1 /flats/ua_09/customer_09_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 14791 15225 -f1 /flats/ua_09/customer_09_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 15226 15660 -f1 /flats/ua_09/customer_09_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 15661 16095 -f1 /flats/ua_10/customer_10_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16096 16530 -f1 /flats/ua_10/customer_10_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16531 16965 -f1 /flats/ua_10/customer_10_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 16966 17400 -f1 /flats/ua_10/customer_10_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 17401 17835 -f1 /flats/ua_11/customer_11_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 17836 18270 -f1 /flats/ua_11/customer_11_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 18271 18705 -f1 /flats/ua_11/customer_11_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 18706 19140 -f1 /flats/ua_11/customer_11_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 19141 19575 -f1 /flats/ua_12/customer_12_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 19576 20010 -f1 /flats/ua_12/customer_12_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20011 20445 -f1 /flats/ua_12/customer_12_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20446 20880 -f1 /flats/ua_12/customer_12_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 20881 21315 -f1 /flats/ua_13/customer_13_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 21316 21750 -f1 /flats/ua_13/customer_13_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 21751 22185 -f1 /flats/ua_13/customer_13_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 22186 22620 -f1 /flats/ua_13/customer_13_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 22621 23055 -f1 /flats/ua_14/customer_14_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 23056 23490 -f1 /flats/ua_14/customer_14_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 23491 23925 -f1 /flats/ua_14/customer_14_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 23926 24360 -f1 /flats/ua_14/customer_14_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 24361 24795 -f1 /flats/ua_15/customer_15_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 24796 25230 -f1 /flats/ua_15/customer_15_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 25231 25665 -f1 /flats/ua_15/customer_15_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 25666 26100 -f1 /flats/ua_15/customer_15_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 26101 26535 -f1 /flats/ua_16/customer_16_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 26536 26970 -f1 /flats/ua_16/customer_16_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 26971 27405 -f1 /flats/ua_16/customer_16_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 27406 27840 -f1 /flats/ua_16/customer_16_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 27841 28275 -f1 /flats/ua_17/customer_17_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 28276 28710 -f1 /flats/ua_17/customer_17_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 28711 29145 -f1 /flats/ua_17/customer_17_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 29146 29580 -f1 /flats/ua_17/customer_17_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 29581 30015 -f1 /flats/ua_18/customer_18_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 30016 30450 -f1 /flats/ua_18/customer_18_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 30451 30885 -f1 /flats/ua_18/customer_18_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 30886 31320 -f1 /flats/ua_18/customer_18_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 31321 31755 -f1 /flats/ua_19/customer_19_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 31756 32190 -f1 /flats/ua_19/customer_19_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 32191 32625 -f1 /flats/ua_19/customer_19_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 32626 33060 -f1 /flats/ua_19/customer_19_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 33061 33495 -f1 /flats/ua_20/customer_20_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 33496 33930 -f1 /flats/ua_20/customer_20_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 33931 34365 -f1 /flats/ua_20/customer_20_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 34366 34800 -f1 /flats/ua_20/customer_20_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 34801 35235 -f1 /flats/ua_21/customer_21_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 35236 35670 -f1 /flats/ua_21/customer_21_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 35671 36105 -f1 /flats/ua_21/customer_21_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 36106 36540 -f1 /flats/ua_21/customer_21_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 36541 36975 -f1 /flats/ua_22/customer_22_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 36976 37410 -f1 /flats/ua_22/customer_22_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 37411 37845 -f1 /flats/ua_22/customer_22_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 37846 38280 -f1 /flats/ua_22/customer_22_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 38281 38715 -f1 /flats/ua_23/customer_23_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 38716 39150 -f1 /flats/ua_23/customer_23_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 39151 39585 -f1 /flats/ua_23/customer_23_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 39586 40020 -f1 /flats/ua_23/customer_23_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 40021 40455 -f1 /flats/ua_24/customer_24_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 40456 40890 -f1 /flats/ua_24/customer_24_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 40891 41325 -f1 /flats/ua_24/customer_24_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 41326 41760 -f1 /flats/ua_24/customer_24_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 41761 42195 -f1 /flats/ua_25/customer_25_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 42196 42630 -f1 /flats/ua_25/customer_25_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 42631 43065 -f1 /flats/ua_25/customer_25_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 43066 43500 -f1 /flats/ua_25/customer_25_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 43501 43935 -f1 /flats/ua_26/customer_26_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 43936 44370 -f1 /flats/ua_26/customer_26_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 44371 44805 -f1 /flats/ua_26/customer_26_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 44806 45240 -f1 /flats/ua_26/customer_26_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 45241 45675 -f1 /flats/ua_27/customer_27_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 45676 46110 -f1 /flats/ua_27/customer_27_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 46111 46545 -f1 /flats/ua_27/customer_27_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 46546 46980 -f1 /flats/ua_27/customer_27_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 46981 47415 -f1 /flats/ua_28/customer_28_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 47416 47850 -f1 /flats/ua_28/customer_28_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 47851 48285 -f1 /flats/ua_28/customer_28_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 48286 48720 -f1 /flats/ua_28/customer_28_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 48721 49155 -f1 /flats/ua_29/customer_29_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 49156 49590 -f1 /flats/ua_29/customer_29_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 49591 50025 -f1 /flats/ua_29/customer_29_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 50026 50460 -f1 /flats/ua_29/customer_29_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 50461 50895 -f1 /flats/ua_30/customer_30_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 50896 51330 -f1 /flats/ua_30/customer_30_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 51331 51765 -f1 /flats/ua_30/customer_30_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 51766 52200 -f1 /flats/ua_30/customer_30_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 52201 52635 -f1 /flats/ua_31/customer_31_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 52636 53070 -f1 /flats/ua_31/customer_31_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 53071 53505 -f1 /flats/ua_31/customer_31_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 53506 53940 -f1 /flats/ua_31/customer_31_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 53941 54375 -f1 /flats/ua_32/customer_32_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 54376 54810 -f1 /flats/ua_32/customer_32_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 54811 55245 -f1 /flats/ua_32/customer_32_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 55246 55680 -f1 /flats/ua_32/customer_32_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 55681 56115 -f1 /flats/ua_33/customer_33_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 56116 56550 -f1 /flats/ua_33/customer_33_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 56551 56985 -f1 /flats/ua_33/customer_33_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 56986 57420 -f1 /flats/ua_33/customer_33_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 57421 57855 -f1 /flats/ua_34/customer_34_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 57856 58290 -f1 /flats/ua_34/customer_34_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 58291 58725 -f1 /flats/ua_34/customer_34_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 58726 59160 -f1 /flats/ua_34/customer_34_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 59161 59595 -f1 /flats/ua_35/customer_35_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 59596 60030 -f1 /flats/ua_35/customer_35_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 60031 60465 -f1 /flats/ua_35/customer_35_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 60466 60900 -f1 /flats/ua_35/customer_35_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 60901 61335 -f1 /flats/ua_36/customer_36_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 61336 61770 -f1 /flats/ua_36/customer_36_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 61771 62205 -f1 /flats/ua_36/customer_36_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 62206 62640 -f1 /flats/ua_36/customer_36_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 62641 63075 -f1 /flats/ua_37/customer_37_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 63076 63510 -f1 /flats/ua_37/customer_37_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 63511 63945 -f1 /flats/ua_37/customer_37_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 63946 64380 -f1 /flats/ua_37/customer_37_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 64381 64815 -f1 /flats/ua_38/customer_38_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 64816 65250 -f1 /flats/ua_38/customer_38_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 65251 65685 -f1 /flats/ua_38/customer_38_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 65686 66120 -f1 /flats/ua_38/customer_38_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 66121 66555 -f1 /flats/ua_39/customer_39_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 66556 66990 -f1 /flats/ua_39/customer_39_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 66991 67425 -f1 /flats/ua_39/customer_39_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 7 -r 67426 67860 -f1 /flats/ua_39/customer_39_4.dat
```

GEN_DISTRICT.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 4 -r 1 67860 -f1 /flats/wdihn/district_1.dat
```

GEN_HISTORY.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 1 16965 -f1 /flats/wdihn/history_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 16966 33930 -f1 /flats/wdihn/history_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 33931 50895 -f1 /flats/wdihn/history_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 8 -r 50896 67860 -f1 /flats/wdihn/history_4.dat
```

GEN_ITEM.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 5 -f1 /flats/wdihn/item_1.dat
```

GEN_NEW_ORDER.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 11 -r 1 67860 -f1 /flats/wdihn/neworder_1.dat
```

GEN_ORDERS.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 1 435 -f1 /flats/ord2/orders_1.dat -f2
/flats/ua_01/orderline_01_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 436 870 -f1 /flats/ord2/orders_2.dat -f2
/flats/ua_01/orderline_01_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 871 1305 -f1 /flats/ord2/orders_3.dat -f2
/flats/ua_01/orderline_01_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 1306 1740 -f1 /flats/ord2/orders_4.dat -f2
/flats/ua_01/orderline_01_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 1741 2175 -f1 /flats/ord2/orders_5.dat -f2
/flats/ua_02/orderline_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 2176 2610 -f1 /flats/ord2/orders_6.dat -f2
/flats/ua_02/orderline_02_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 2611 3045 -f1 /flats/ord2/orders_7.dat -f2
/flats/ua_02/orderline_02_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 3046 3480 -f1 /flats/ord2/orders_8.dat -f2
/flats/ua_02/orderline_02_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 3481 3915 -f1 /flats/ord2/orders_9.dat -f2
/flats/ua_03/orderline_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 3916 4350 -f1 /flats/ord2/orders_10.dat -f2
/flats/ua_03/orderline_03_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 4351 4785 -f1 /flats/ord2/orders_11.dat -f2
/flats/ua_03/orderline_03_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 4786 5220 -f1 /flats/ord2/orders_12.dat -f2
/flats/ua_03/orderline_03_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 5221 5655 -f1 /flats/ord2/orders_13.dat -f2
/flats/ua_04/orderline_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 5656 6090 -f1 /flats/ord2/orders_14.dat -f2
/flats/ua_04/orderline_04_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 6091 6525 -f1 /flats/ord2/orders_15.dat -f2
/flats/ua_04/orderline_04_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 6526 6960 -f1 /flats/ord2/orders_16.dat -f2
/flats/ua_04/orderline_04_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 6961 7395 -f1 /flats/ord2/orders_17.dat -f2
/flats/ua_05/orderline_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 7396 7830 -f1 /flats/ord2/orders_18.dat -f2
/flats/ua_05/orderline_05_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 7831 8265 -f1 /flats/ord2/orders_19.dat -f2
/flats/ua_05/orderline_05_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 8266 8700 -f1 /flats/ord2/orders_20.dat -f2
/flats/ua_05/orderline_05_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 8701 9135 -f1 /flats/ord2/orders_21.dat -f2
/flats/ua_06/orderline_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 9136 9570 -f1 /flats/ord2/orders_22.dat -f2
/flats/ua_06/orderline_06_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 9571 10005 -f1 /flats/ord2/orders_23.dat -f2
/flats/ua_06/orderline_06_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 10006 10440 -f1 /flats/ord2/orders_24.dat -f2
/flats/ua_06/orderline_06_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 10441 10875 -f1 /flats/ord2/orders_25.dat -f2
/flats/ua_07/orderline_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 10876 11310 -f1 /flats/ord2/orders_26.dat -f2
/flats/ua_07/orderline_07_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 11311 11745 -f1 /flats/ord2/orders_27.dat -f2
/flats/ua_07/orderline_07_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 11746 12180 -f1 /flats/ord2/orders_28.dat -f2
/flats/ua_07/orderline_07_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 12181 12615 -f1 /flats/ord2/orders_29.dat -f2
/flats/ua_08/orderline_08_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 12616 13050 -f1 /flats/ord2/orders_30.dat -f2
/flats/ua_08/orderline_08_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 13051 13485 -f1 /flats/ord2/orders_31.dat -f2
/flats/ua_08/orderline_08_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 13486 13920 -f1 /flats/ord2/orders_32.dat -f2
/flats/ua_08/orderline_08_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 13921 14355 -f1 /flats/ord2/orders_33.dat -f2
/flats/ua_09/orderline_09_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 14356 14790 -f1 /flats/ord2/orders_34.dat -f2
/flats/ua_09/orderline_09_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 14791 15225 -f1 /flats/ord2/orders_35.dat -f2
/flats/ua_09/orderline_09_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 15226 15660 -f1 /flats/ord2/orders_36.dat -f2
/flats/ua_09/orderline_09_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 15661 16095 -f1 /flats/ord2/orders_37.dat -f2
/flats/ua_10/orderline_10_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 16096 16530 -f1 /flats/ord2/orders_38.dat -f2
/flats/ua_10/orderline_10_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 16531 16965 -f1 /flats/ord2/orders_39.dat -f2
/flats/ua_10/orderline_10_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 16966 17400 -f1 /flats/ord2/orders_40.dat -f2
/flats/ua_10/orderline_10_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 17401 17835 -f1 /flats/ord2/orders_41.dat -f2
/flats/ua_11/orderline_11_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 17836 18270 -f1 /flats/ord2/orders_42.dat -f2
/flats/ua_11/orderline_11_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 18271 18705 -f1 /flats/ord2/orders_43.dat -f2
/flats/ua_11/orderline_11_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 18706 19140 -f1 /flats/ord2/orders_44.dat -f2
/flats/ua_11/orderline_11_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 19141 19575 -f1 /flats/ord2/orders_45.dat -f2
/flats/ua_12/orderline_12_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 19576 20010 -f1 /flats/ord2/orders_46.dat -f2
/flats/ua_12/orderline_12_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 20011 20445 -f1 /flats/ord2/orders_47.dat -f2
/flats/ua_12/orderline_12_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 20446 20880 -f1 /flats/ord2/orders_48.dat -f2
/flats/ua_12/orderline_12_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 20881 21315 -f1 /flats/ord2/orders_49.dat -f2
/flats/ua_13/orderline_13_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 21316 21750 -f1 /flats/ord2/orders_50.dat -f2
/flats/ua_13/orderline_13_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 21751 22185 -f1 /flats/ord2/orders_51.dat -f2
/flats/ua_13/orderline_13_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 22186 22620 -f1 /flats/ord2/orders_52.dat -f2
/flats/ua_13/orderline_13_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 22621 23055 -f1 /flats/ord2/orders_53.dat -f2
/flats/ua_14/orderline_14_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 23056 23490 -f1 /flats/ord2/orders_54.dat -f2
/flats/ua_14/orderline_14_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 23491 23925 -f1 /flats/ord2/orders_55.dat -f2
/flats/ua_14/orderline_14_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 23926 24360 -f1 /flats/ord2/orders_56.dat -f2
/flats/ua_14/orderline_14_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 24361 24795 -f1 /flats/ord2/orders_57.dat -f2
/flats/ua_15/orderline_15_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 24796 25230 -f1 /flats/ord2/orders_58.dat -f2
/flats/ua_15/orderline_15_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 25231 25665 -f1 /flats/ord2/orders_59.dat -f2
/flats/ua_15/orderline_15_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 25666 26100 -f1 /flats/ord2/orders_60.dat -f2
/flats/ua_15/orderline_15_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 26101 26535 -f1 /flats/ord2/orders_61.dat -f2
/flats/ua_16/orderline_16_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 26536 26970 -f1 /flats/ord2/orders_62.dat -f2
/flats/ua_16/orderline_16_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 26971 27405 -f1 /flats/ord2/orders_63.dat -f2
/flats/ua_16/orderline_16_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 27406 27840 -f1 /flats/ord2/orders_64.dat -f2
/flats/ua_16/orderline_16_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 27841 28275 -f1 /flats/ord2/orders_65.dat -f2
/flats/ua_17/orderline_17_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 28276 28710 -f1 /flats/ord2/orders_66.dat -f2
/flats/ua_17/orderline_17_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 28711 29145 -f1 /flats/ord2/orders_67.dat -f2
/flats/ua_17/orderline_17_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 29146 29580 -f1 /flats/ord2/orders_68.dat -f2
/flats/ua_17/orderline_17_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 29581 30015 -f1 /flats/ord2/orders_69.dat -f2
/flats/ua_18/orderline_18_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 30016 30450 -f1 /flats/ord2/orders_70.dat -f2
/flats/ua_18/orderline_18_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 30451 30885 -f1 /flats/ord2/orders_71.dat -f2
/flats/ua_18/orderline_18_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 30886 31320 -f1 /flats/ord2/orders_72.dat -f2
/flats/ua_18/orderline_18_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 31321 31755 -f1 /flats/ord2/orders_73.dat -f2
/flats/ua_19/orderline_19_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 31756 32190 -f1 /flats/ord2/orders_74.dat -f2
/flats/ua_19/orderline_19_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 32191 32625 -f1 /flats/ord2/orders_75.dat -f2
/flats/ua_19/orderline_19_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 32626 33060 -f1 /flats/ord2/orders_76.dat -f2
/flats/ua_19/orderline_19_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 33061 33495 -f1 /flats/ord2/orders_77.dat -f2
/flats/ua_20/orderline_20_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 33496 33930 -f1 /flats/ord2/orders_78.dat -f2
/flats/ua_20/orderline_20_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 33931 34365 -f1 /flats/ord2/orders_79.dat -f2
/flats/ua_20/orderline_20_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 34366 34800 -f1 /flats/ord2/orders_80.dat -f2
/flats/ua_20/orderline_20_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 34801 35235 -f1 /flats/ord2/orders_81.dat -f2
/flats/ua_21/orderline_21_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 35236 35670 -f1 /flats/ord2/orders_82.dat -f2
/flats/ua_21/orderline_21_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 35671 36105 -f1 /flats/ord2/orders_83.dat -f2
/flats/ua_21/orderline_21_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 36106 36540 -f1 /flats/ord2/orders_84.dat -f2
/flats/ua_21/orderline_21_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 36541 36975 -f1 /flats/ord2/orders_85.dat -f2
/flats/ua_22/orderline_22_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 36976 37410 -f1 /flats/ord2/orders_86.dat -f2
/flats/ua_22/orderline_22_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 37411 37845 -f1 /flats/ord2/orders_87.dat -f2
/flats/ua_22/orderline_22_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 37846 38280 -f1 /flats/ord2/orders_88.dat -f2
/flats/ua_22/orderline_22_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 38281 38715 -f1 /flats/ord2/orders_89.dat -f2
/flats/ua_23/orderline_23_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 38716 39150 -f1 /flats/ord2/orders_90.dat -f2
/flats/ua_23/orderline_23_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 39151 39585 -f1 /flats/ord2/orders_91.dat -f2
/flats/ua_23/orderline_23_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 39586 40020 -f1 /flats/ord2/orders_92.dat -f2
/flats/ua_23/orderline_23_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 40021 40455 -f1 /flats/ord2/orders_93.dat -f2
```

```
/flats/ua_24/orderline_24_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 40456 40890 -f1 /flats/ord2/orders_94.dat -f2
/flats/ua_24/orderline_24_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 40891 41325 -f1 /flats/ord2/orders_95.dat -f2
/flats/ua_24/orderline_24_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 41326 41760 -f1 /flats/ord2/orders_96.dat -f2
/flats/ua_24/orderline_24_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 41761 42195 -f1 /flats/ord2/orders_97.dat -f2
/flats/ua_25/orderline_25_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 42196 42630 -f1 /flats/ord2/orders_98.dat -f2
/flats/ua_25/orderline_25_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 42631 43065 -f1 /flats/ord2/orders_99.dat -f2
/flats/ua_25/orderline_25_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 43066 43500 -f1 /flats/ord2/orders_100.dat -f2
/flats/ua_25/orderline_25_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 43501 43935 -f1 /flats/ord2/orders_101.dat -f2
/flats/ua_26/orderline_26_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 43936 44370 -f1 /flats/ord2/orders_102.dat -f2
/flats/ua_26/orderline_26_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 44371 44805 -f1 /flats/ord2/orders_103.dat -f2
/flats/ua_26/orderline_26_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 44806 45240 -f1 /flats/ord2/orders_104.dat -f2
/flats/ua_26/orderline_26_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 45241 45675 -f1 /flats/ord2/orders_105.dat -f2
/flats/ua_27/orderline_27_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 45676 46110 -f1 /flats/ord2/orders_106.dat -f2
/flats/ua_27/orderline_27_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 46111 46545 -f1 /flats/ord2/orders_107.dat -f2
/flats/ua_27/orderline_27_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 46546 46980 -f1 /flats/ord2/orders_108.dat -f2
/flats/ua_27/orderline_27_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 46981 47415 -f1 /flats/ord2/orders_109.dat -f2
/flats/ua_28/orderline_28_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 47416 47850 -f1 /flats/ord2/orders_110.dat -f2
/flats/ua_28/orderline_28_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 47851 48285 -f1 /flats/ord2/orders_111.dat -f2
/flats/ua_28/orderline_28_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 48286 48720 -f1 /flats/ord2/orders_112.dat -f2
/flats/ua_28/orderline_28_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 48721 49155 -f1 /flats/ord2/orders_113.dat -f2
/flats/ua_29/orderline_29_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 49156 49590 -f1 /flats/ord2/orders_114.dat -f2
/flats/ua_29/orderline_29_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 49591 50025 -f1 /flats/ord2/orders_115.dat -f2
/flats/ua_29/orderline_29_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 50026 50460 -f1 /flats/ord2/orders_116.dat -f2
/flats/ua_29/orderline_29_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 50461 50895 -f1 /flats/ord2/orders_117.dat -f2
/flats/ua_30/orderline_30_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 50896 51330 -f1 /flats/ord2/orders_118.dat -f2
/flats/ua_30/orderline_30_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 51331 51765 -f1 /flats/ord2/orders_119.dat -f2
/flats/ua_30/orderline_30_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 51766 52200 -f1 /flats/ord2/orders_120.dat -f2
/flats/ua_30/orderline_30_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 52201 52635 -f1 /flats/ord2/orders_121.dat -f2
/flats/ua_31/orderline_31_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 52636 53070 -f1 /flats/ord2/orders_122.dat -f2
/flats/ua_31/orderline_31_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 53071 53505 -f1 /flats/ord2/orders_123.dat -f2
/flats/ua_31/orderline_31_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 53506 53940 -f1 /flats/ord2/orders_124.dat -f2
/flats/ua_31/orderline_31_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 53941 54375 -f1 /flats/ord2/orders_125.dat -f2
/flats/ua_32/orderline_32_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 54376 54810 -f1 /flats/ord2/orders_126.dat -f2
/flats/ua_32/orderline_32_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 54811 55245 -f1 /flats/ord2/orders_127.dat -f2
/flats/ua_32/orderline_32_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 55246 55680 -f1 /flats/ord2/orders_128.dat -f2
/flats/ua_32/orderline_32_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 55681 56115 -f1 /flats/ord2/orders_129.dat -f2
/flats/ua_33/orderline_33_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 56116 56550 -f1 /flats/ord2/orders_130.dat -f2
/flats/ua_33/orderline_33_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 56551 56985 -f1 /flats/ord2/orders_131.dat -f2
/flats/ua_33/orderline_33_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 56986 57420 -f1 /flats/ord2/orders_132.dat -f2
/flats/ua_33/orderline_33_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 57421 57855 -f1 /flats/ord2/orders_133.dat -f2
/flats/ua_34/orderline_34_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 57856 58290 -f1 /flats/ord2/orders_134.dat -f2
/flats/ua_34/orderline_34_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 58291 58725 -f1 /flats/ord2/orders_135.dat -f2
/flats/ua_34/orderline_34_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 58726 59160 -f1 /flats/ord2/orders_136.dat -f2
/flats/ua_34/orderline_34_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 59161 59595 -f1 /flats/ord2/orders_137.dat -f2
/flats/ua_35/orderline_35_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 59596 60030 -f1 /flats/ord2/orders_138.dat -f2
/flats/ua_35/orderline_35_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 60031 60465 -f1 /flats/ord2/orders_139.dat -f2
/flats/ua_35/orderline_35_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 60466 60900 -f1 /flats/ord2/orders_140.dat -f2
/flats/ua_35/orderline_35_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 60901 61335 -f1 /flats/ord2/orders_141.dat -f2
/flats/ua_36/orderline_36_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 61336 61770 -f1 /flats/ord2/orders_142.dat -f2
/flats/ua_36/orderline_36_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 61771 62205 -f1 /flats/ord2/orders_143.dat -f2
/flats/ua_36/orderline_36_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 62206 62640 -f1 /flats/ord2/orders_144.dat -f2
/flats/ua_36/orderline_36_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 62641 63075 -f1 /flats/ord2/orders_145.dat -f2
/flats/ua_37/orderline_37_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 63076 63510 -f1 /flats/ord2/orders_146.dat -f2
/flats/ua_37/orderline_37_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 63511 63945 -f1 /flats/ord2/orders_147.dat -f2
/flats/ua_37/orderline_37_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 63946 64380 -f1 /flats/ord2/orders_148.dat -f2
/flats/ua_37/orderline_37_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 64381 64815 -f1 /flats/ord2/orders_149.dat -f2
/flats/ua_38/orderline_38_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 64816 65250 -f1 /flats/ord2/orders_150.dat -f2
/flats/ua_38/orderline_38_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 65251 65685 -f1 /flats/ord2/orders_151.dat -f2
/flats/ua_38/orderline_38_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 65686 66120 -f1 /flats/ord2/orders_152.dat -f2
/flats/ua_38/orderline_38_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 66121 66555 -f1 /flats/ord2/orders_153.dat -f2
/flats/ua_39/orderline_39_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 66556 66990 -f1 /flats/ord2/orders_154.dat -f2
/flats/ua_39/orderline_39_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 66991 67425 -f1 /flats/ord2/orders_155.dat -f2
/flats/ua_39/orderline_39_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 9 -r 67426 67860 -f1 /flats/ord2/orders_156.dat -f2
/flats/ua_39/orderline_39_4.dat
```

GEN_STOCK.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1 435 -f1 /flats/ua_01/stock_01_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 436 870 -f1 /flats/ua_01/stock_01_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 871 1305 -f1 /flats/ua_01/stock_01_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1306 1740 -f1 /flats/ua_01/stock_01_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 1741 2175 -f1 /flats/ua_02/stock_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 2176 2610 -f1 /flats/ua_02/stock_02_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 2611 3045 -f1 /flats/ua_02/stock_02_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 3046 3480 -f1 /flats/ua_02/stock_02_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 3481 3915 -f1 /flats/ua_03/stock_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 3916 4350 -f1 /flats/ua_03/stock_03_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 4351 4785 -f1 /flats/ua_03/stock_03_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 4786 5220 -f1 /flats/ua_03/stock_03_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 5221 5655 -f1 /flats/ua_04/stock_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 5656 6090 -f1 /flats/ua_04/stock_04_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 6091 6525 -f1 /flats/ua_04/stock_04_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 6526 6960 -f1 /flats/ua_04/stock_04_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 6961 7395 -f1 /flats/ua_05/stock_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 7396 7830 -f1 /flats/ua_05/stock_05_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 7831 8265 -f1 /flats/ua_05/stock_05_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 8266 8700 -f1 /flats/ua_05/stock_05_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 8701 9135 -f1 /flats/ua_06/stock_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 9136 9570 -f1 /flats/ua_06/stock_06_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 9571 10005 -f1 /flats/ua_06/stock_06_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 10006 10440 -f1 /flats/ua_06/stock_06_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 10441 10875 -f1 /flats/ua_07/stock_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 10876 11310 -f1 /flats/ua_07/stock_07_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 11311 11745 -f1 /flats/ua_07/stock_07_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 11746 12180 -f1 /flats/ua_07/stock_07_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 12181 12615 -f1 /flats/ua_08/stock_08_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 12616 13050 -f1 /flats/ua_08/stock_08_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 13051 13485 -f1 /flats/ua_08/stock_08_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 13486 13920 -f1 /flats/ua_08/stock_08_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 13921 14355 -f1 /flats/ua_09/stock_09_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 14356 14790 -f1 /flats/ua_09/stock_09_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 14791 15225 -f1 /flats/ua_09/stock_09_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 15226 15660 -f1 /flats/ua_09/stock_09_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 15661 16095 -f1 /flats/ua_10/stock_10_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 16096 16530 -f1 /flats/ua_10/stock_10_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 16531 16965 -f1 /flats/ua_10/stock_10_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 16966 17400 -f1 /flats/ua_10/stock_10_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 17401 17835 -f1 /flats/ua_11/stock_11_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 17836 18270 -f1 /flats/ua_11/stock_11_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 18271 18705 -f1 /flats/ua_11/stock_11_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 18706 19140 -f1 /flats/ua_11/stock_11_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 19141 19575 -f1 /flats/ua_12/stock_12_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 19576 20010 -f1 /flats/ua_12/stock_12_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 20011 20445 -f1 /flats/ua_12/stock_12_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 20446 20880 -f1 /flats/ua_12/stock_12_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 20881 21315 -f1 /flats/ua_13/stock_13_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 21316 21750 -f1 /flats/ua_13/stock_13_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 21751 22185 -f1 /flats/ua_13/stock_13_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 22186 22620 -f1 /flats/ua_13/stock_13_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 22621 23055 -f1 /flats/ua_14/stock_14_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 23056 23490 -f1 /flats/ua_14/stock_14_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 23491 23925 -f1 /flats/ua_14/stock_14_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 23926 24360 -f1 /flats/ua_14/stock_14_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 24361 24795 -f1 /flats/ua_15/stock_15_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 24796 25230 -f1 /flats/ua_15/stock_15_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 25231 25665 -f1 /flats/ua_15/stock_15_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 25666 26100 -f1 /flats/ua_15/stock_15_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 26101 26535 -f1 /flats/ua_16/stock_16_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 26536 26970 -f1 /flats/ua_16/stock_16_2.dat
```

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 26971 27405 -f1 /flats/ua_16/stock_16_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 27406 27840 -f1 /flats/ua_16/stock_16_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 27841 28275 -f1 /flats/ua_17/stock_17_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 28276 28710 -f1 /flats/ua_17/stock_17_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 28711 29145 -f1 /flats/ua_17/stock_17_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 29146 29580 -f1 /flats/ua_17/stock_17_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 29581 30015 -f1 /flats/ua_18/stock_18_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 30016 30450 -f1 /flats/ua_18/stock_18_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 30451 30885 -f1 /flats/ua_18/stock_18_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 30886 31320 -f1 /flats/ua_18/stock_18_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 31321 31755 -f1 /flats/ua_19/stock_19_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 31756 32190 -f1 /flats/ua_19/stock_19_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 32191 32625 -f1 /flats/ua_19/stock_19_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 32626 33060 -f1 /flats/ua_19/stock_19_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 33061 33495 -f1 /flats/ua_20/stock_20_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 33496 33930 -f1 /flats/ua_20/stock_20_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 33931 34365 -f1 /flats/ua_20/stock_20_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 34366 34800 -f1 /flats/ua_20/stock_20_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 34801 35235 -f1 /flats/ua_21/stock_21_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 35236 35670 -f1 /flats/ua_21/stock_21_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 35671 36105 -f1 /flats/ua_21/stock_21_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 36106 36540 -f1 /flats/ua_21/stock_21_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 36541 36975 -f1 /flats/ua_22/stock_22_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 36976 37410 -f1 /flats/ua_22/stock_22_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 37411 37845 -f1 /flats/ua_22/stock_22_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 37846 38280 -f1 /flats/ua_22/stock_22_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 38281 38715 -f1 /flats/ua_23/stock_23_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 38716 39150 -f1 /flats/ua_23/stock_23_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 39151 39585 -f1 /flats/ua_23/stock_23_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 39586 40020 -f1 /flats/ua_23/stock_23_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 40021 40455 -f1 /flats/ua_24/stock_24_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 40456 40890 -f1 /flats/ua_24/stock_24_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 40891 41325 -f1 /flats/ua_24/stock_24_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 41326 41760 -f1 /flats/ua_24/stock_24_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 41761 42195 -f1 /flats/ua_25/stock_25_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 42196 42630 -f1 /flats/ua_25/stock_25_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 42631 43065 -f1 /flats/ua_25/stock_25_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 43066 43500 -f1 /flats/ua_25/stock_25_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 43501 43935 -f1 /flats/ua_26/stock_26_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 43936 44370 -f1 /flats/ua_26/stock_26_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 44371 44805 -f1 /flats/ua_26/stock_26_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 44806 45240 -f1 /flats/ua_26/stock_26_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 45241 45675 -f1 /flats/ua_27/stock_27_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 45676 46110 -f1 /flats/ua_27/stock_27_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 46111 46545 -f1 /flats/ua_27/stock_27_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 46546 46980 -f1 /flats/ua_27/stock_27_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 46981 47415 -f1 /flats/ua_28/stock_28_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 47416 47850 -f1 /flats/ua_28/stock_28_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 47851 48285 -f1 /flats/ua_28/stock_28_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 48286 48720 -f1 /flats/ua_28/stock_28_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 48721 49155 -f1 /flats/ua_29/stock_29_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 49156 49590 -f1 /flats/ua_29/stock_29_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 49591 50025 -f1 /flats/ua_29/stock_29_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 50026 50460 -f1 /flats/ua_29/stock_29_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 50461 50895 -f1 /flats/ua_30/stock_30_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 50896 51330 -f1 /flats/ua_30/stock_30_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 51331 51765 -f1 /flats/ua_30/stock_30_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 51766 52200 -f1 /flats/ua_30/stock_30_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 52201 52635 -f1 /flats/ua_31/stock_31_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 52636 53070 -f1 /flats/ua_31/stock_31_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 53071 53505 -f1 /flats/ua_31/stock_31_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 53506 53940 -f1 /flats/ua_31/stock_31_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 53941 54375 -f1 /flats/ua_32/stock_32_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 54376 54810 -f1 /flats/ua_32/stock_32_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 54811 55245 -f1 /flats/ua_32/stock_32_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 55246 55680 -f1 /flats/ua_32/stock_32_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 55681 56115 -f1 /flats/ua_33/stock_33_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 56116 56550 -f1 /flats/ua_33/stock_33_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 56551 56985 -f1 /flats/ua_33/stock_33_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 56986 57420 -f1 /flats/ua_33/stock_33_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 57421 57855 -f1 /flats/ua_34/stock_34_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 57856 58290 -f1 /flats/ua_34/stock_34_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 58291 58725 -f1 /flats/ua_34/stock_34_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 58726 59160 -f1 /flats/ua_34/stock_34_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 59161 59595 -f1 /flats/ua_35/stock_35_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 59596 60030 -f1 /flats/ua_35/stock_35_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 60031 60465 -f1 /flats/ua_35/stock_35_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 60466 60900 -f1 /flats/ua_35/stock_35_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 60901 61335 -f1 /flats/ua_36/stock_36_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 61336 61770 -f1 /flats/ua_36/stock_36_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 61771 62205 -f1 /flats/ua_36/stock_36_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 62206 62640 -f1 /flats/ua_36/stock_36_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 62641 63075 -f1 /flats/ua_37/stock_37_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 63076 63510 -f1 /flats/ua_37/stock_37_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 63511 63945 -f1 /flats/ua_37/stock_37_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 63946 64380 -f1 /flats/ua_37/stock_37_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 64381 64815 -f1 /flats/ua_38/stock_38_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 64816 65250 -f1 /flats/ua_38/stock_38_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 65251 65685 -f1 /flats/ua_38/stock_38_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 65686 66120 -f1 /flats/ua_38/stock_38_4.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 66121 66555 -f1 /flats/ua_39/stock_39_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 66556 66990 -f1 /flats/ua_39/stock_39_2.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 66991 67425 -f1 /flats/ua_39/stock_39_3.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 6 -r 67426 67860 -f1 /flats/ua_39/stock_39_4.dat
```

GEN_WAREHOUSE.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t 3 -r 1 67860 -f1 /flats/wdihn/warehouse_1.dat
```

LOAD_CUSTOMER_ALL.ddl

```
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/customer_01_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/customer_01_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/customer_01_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/customer_01_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_02/customer_02_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_02/customer_02_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_02/customer_02_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_02/customer_02_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_03/customer_03_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_03/customer_03_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_03/customer_03_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_03/customer_03_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_04/customer_04_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_04/customer_04_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_04/customer_04_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER4;
```

```
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_04/customer_04_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_05/customer_05_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_05/customer_05_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_05/customer_05_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_05/customer_05_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_06/customer_06_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_06/customer_06_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_06/customer_06_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_06/customer_06_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_07/customer_07_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_07/customer_07_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_07/customer_07_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_07/customer_07_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_08/customer_08_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_08/customer_08_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_08/customer_08_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_08/customer_08_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_09/customer_09_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_09/customer_09_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_09/customer_09_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_09/customer_09_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_10/customer_10_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_10/customer_10_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_10/customer_10_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_10/customer_10_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_11/customer_11_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_11/customer_11_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_11/customer_11_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_11/customer_11_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_12/customer_12_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_12/customer_12_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_12/customer_12_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_12/customer_12_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER12;
```

```
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_13/customer_13_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_13/customer_13_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_13/customer_13_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_13/customer_13_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_14/customer_14_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_14/customer_14_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_14/customer_14_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_14/customer_14_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_15/customer_15_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_15/customer_15_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_15/customer_15_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_15/customer_15_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_16/customer_16_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_16/customer_16_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_16/customer_16_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_16/customer_16_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_17/customer_17_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_17/customer_17_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_17/customer_17_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_17/customer_17_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_18/customer_18_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_18/customer_18_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_18/customer_18_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_18/customer_18_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_19/customer_19_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_19/customer_19_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_19/customer_19_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_19/customer_19_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_20/customer_20_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_20/customer_20_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_20/customer_20_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_20/customer_20_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_21/customer_21_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER21;
```

```
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_21/customer_21_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_21/customer_21_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_21/customer_21_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_22/customer_22_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_22/customer_22_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_22/customer_22_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_22/customer_22_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_23/customer_23_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_23/customer_23_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_23/customer_23_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_23/customer_23_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_24/customer_24_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_24/customer_24_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_24/customer_24_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_24/customer_24_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_25/customer_25_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_25/customer_25_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_25/customer_25_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_25/customer_25_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_26/customer_26_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_26/customer_26_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_26/customer_26_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_26/customer_26_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_27/customer_27_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_27/customer_27_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_27/customer_27_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_27/customer_27_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_28/customer_28_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_28/customer_28_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_28/customer_28_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_28/customer_28_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_29/customer_29_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_29/customer_29_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER29;
```

```
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_29/customer_29_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_29/customer_29_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_30/customer_30_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_30/customer_30_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_30/customer_30_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_30/customer_30_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/customer_31_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/customer_31_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/customer_31_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/customer_31_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/customer_32_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/customer_32_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/customer_32_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/customer_32_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/customer_33_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/customer_33_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/customer_33_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/customer_33_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/customer_34_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/customer_34_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/customer_34_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/customer_34_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/customer_35_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/customer_35_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/customer_35_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/customer_35_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/customer_36_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/customer_36_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/customer_36_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/customer_36_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/customer_37_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/customer_37_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/customer_37_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER37;
```

```
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/customer_37_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/customer_38_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/customer_38_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/customer_38_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/customer_38_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_39/customer_39_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_39/customer_39_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_39/customer_39_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE CUSTOMER39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_39/customer_39_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 13500000 INSERT INTO CUSTOMER39;
COMMIT WORK;
CONNECT RESET;
```

LOAD_DISTRICT_ALL.ddl

```
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/wdihn/district_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 1000 INSERT INTO DISTRICT;
COMMIT WORK;
CONNECT RESET;
```

LOAD_HISTORY_ALL.ddl

```
connect to tpcc in share mode;
LOAD FROM /flats/wdihn/history_1.dat, /flats/wdihn/history_2.dat, /flats/wdihn/history_3.dat,
/flats/wdihn/history_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS FASTPARSE REPLACE
INTO HISTORY NONRECOVERABLE DATA BUFFER 16000 CPU_PARALLELISM 4 ;
connect reset;
```

LOAD_ITEM_ALL.ddl

```
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/wdihn/item_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 1000 INSERT INTO ITEM;
COMMIT WORK;
CONNECT RESET;
```

LOAD_NEW-ORDER_ALL.ddl

```
connect to tpcc in share mode;
LOAD FROM /flats/wdihn/neworder_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
FASTPARSE REPLACE INTO NEW_ORDER NONRECOVERABLE DATA BUFFER 16000
CPU_PARALLELISM 4 ;
connect reset;
```

LOAD_ORDERS_ALL.ddl

```
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_5.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_6.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_7.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_8.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_9.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_10.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_11.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_12.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_13.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_14.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_15.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_16.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_17.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
```

```
IMPORT FROM /flats/ord2/orders_18.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_19.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_20.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_21.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_22.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_23.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_24.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_25.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_26.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_27.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_28.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_29.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_30.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_31.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_32.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_33.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_34.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_35.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_36.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_37.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_38.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_39.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_40.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_41.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_42.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_43.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_44.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_45.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_46.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_47.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_48.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_49.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_50.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_51.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_52.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_53.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_54.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_55.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_56.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_57.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_58.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_59.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_60.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_61.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_62.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_63.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_64.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_65.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_66.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_67.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_68.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_69.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_70.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_71.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_72.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_73.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_74.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_75.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_76.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_77.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_78.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_79.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_80.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_81.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_82.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_83.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_84.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_85.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_86.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_87.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_88.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_89.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_90.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_91.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_92.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_93.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_94.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_95.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_96.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_97.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
```

```
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_98.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_99.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_100.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_101.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_102.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_103.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_104.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_105.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_106.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_107.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_108.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_109.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_110.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_111.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_112.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_113.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_114.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_115.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_116.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_117.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_118.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_119.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_120.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_121.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_122.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_123.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_124.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_125.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_126.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_127.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_128.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_129.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_130.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_131.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_132.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_133.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_134.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_135.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_136.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
```

```
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_137.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_138.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_139.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_140.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_141.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_142.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_143.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_144.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_145.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_146.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_147.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_148.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_149.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_150.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_151.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_152.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_153.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_154.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_155.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 INSERT INTO ORDERS;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/ord2/orders_156.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 50000 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 /flats/ua_01/orderline_01_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/orderline_01_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/orderline_01_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/orderline_01_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_02/orderline_02_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_02/orderline_02_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_02/orderline_02_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_02/orderline_02_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_03/orderline_03_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_03/orderline_03_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_03/orderline_03_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_03/orderline_03_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_04/orderline_04_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_04/orderline_04_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_04/orderline_04_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE4;
COMMIT WORK;
CONNECT RESET;
```

```
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_04/orderline_04_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_05/orderline_05_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_05/orderline_05_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_05/orderline_05_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_05/orderline_05_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_06/orderline_06_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_06/orderline_06_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_06/orderline_06_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_06/orderline_06_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_07/orderline_07_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_07/orderline_07_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_07/orderline_07_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_07/orderline_07_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_08/orderline_08_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_08/orderline_08_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_08/orderline_08_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_08/orderline_08_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_09/orderline_09_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_09/orderline_09_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_09/orderline_09_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_09/orderline_09_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_10/orderline_10_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_10/orderline_10_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_10/orderline_10_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_10/orderline_10_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_11/orderline_11_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_11/orderline_11_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_11/orderline_11_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_11/orderline_11_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_12/orderline_12_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_12/orderline_12_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_12/orderline_12_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_12/orderline_12_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE12;
COMMIT WORK;
CONNECT RESET;
```

```
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_13/orderline_13_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_13/orderline_13_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_13/orderline_13_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_13/orderline_13_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_14/orderline_14_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_14/orderline_14_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_14/orderline_14_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_14/orderline_14_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_15/orderline_15_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_15/orderline_15_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_15/orderline_15_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_15/orderline_15_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_16/orderline_16_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_16/orderline_16_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_16/orderline_16_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_16/orderline_16_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_17/orderline_17_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_17/orderline_17_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_17/orderline_17_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_17/orderline_17_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_18/orderline_18_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_18/orderline_18_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_18/orderline_18_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_18/orderline_18_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_19/orderline_19_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_19/orderline_19_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_19/orderline_19_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_19/orderline_19_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_20/orderline_20_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_20/orderline_20_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_20/orderline_20_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_20/orderline_20_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_21/orderline_21_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE21;
COMMIT WORK;
CONNECT RESET;
```

```
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_21/orderline_21_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_21/orderline_21_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_21/orderline_21_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_22/orderline_22_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_22/orderline_22_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_22/orderline_22_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_22/orderline_22_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_23/orderline_23_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_23/orderline_23_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_23/orderline_23_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_23/orderline_23_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_24/orderline_24_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_24/orderline_24_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_24/orderline_24_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_24/orderline_24_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_25/orderline_25_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_25/orderline_25_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_25/orderline_25_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_25/orderline_25_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_26/orderline_26_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_26/orderline_26_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_26/orderline_26_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_26/orderline_26_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_27/orderline_27_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_27/orderline_27_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_27/orderline_27_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_27/orderline_27_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_28/orderline_28_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_28/orderline_28_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_28/orderline_28_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_28/orderline_28_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_29/orderline_29_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_29/orderline_29_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE29;
COMMIT WORK;
CONNECT RESET;
```

```
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_29/orderline_29_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_29/orderline_29_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_30/orderline_30_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_30/orderline_30_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_30/orderline_30_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_30/orderline_30_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/orderline_31_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/orderline_31_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/orderline_31_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/orderline_31_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/orderline_32_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/orderline_32_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/orderline_32_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/orderline_32_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/orderline_33_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/orderline_33_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/orderline_33_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/orderline_33_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/orderline_34_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/orderline_34_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/orderline_34_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/orderline_34_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/orderline_35_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/orderline_35_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/orderline_35_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/orderline_35_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/orderline_36_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/orderline_36_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/orderline_36_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/orderline_36_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/orderline_37_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/orderline_37_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/orderline_37_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE37;
COMMIT WORK;
CONNECT RESET;
```

```
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/orderline_37_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/orderline_38_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/orderline_38_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/orderline_38_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/orderline_38_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_39/orderline_39_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_39/orderline_39_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_39/orderline_39_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE ORDER_LINE39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_39/orderline_39_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 157000000 INSERT INTO ORDER_LINE39;
COMMIT WORK;
CONNECT RESET;
```

LOAD_STOCK_ALL.ddl

```
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/stock_01_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/stock_01_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/stock_01_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK1 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_01/stock_01_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_02/stock_02_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_02/stock_02_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_02/stock_02_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK2 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_02/stock_02_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_03/stock_03_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_03/stock_03_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_03/stock_03_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK3 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_03/stock_03_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_04/stock_04_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_04/stock_04_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_04/stock_04_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK4 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_04/stock_04_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_05/stock_05_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_05/stock_05_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_05/stock_05_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK5 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_05/stock_05_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_06/stock_06_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_06/stock_06_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_06/stock_06_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK6;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK6 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_06/stock_06_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_07/stock_07_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_07/stock_07_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_07/stock_07_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK7 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_07/stock_07_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_08/stock_08_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_08/stock_08_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_08/stock_08_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK8 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_08/stock_08_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_09/stock_09_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_09/stock_09_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_09/stock_09_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK9 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_09/stock_09_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_10/stock_10_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_10/stock_10_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_10/stock_10_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK10 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_10/stock_10_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK10;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_11/stock_11_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_11/stock_11_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_11/stock_11_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK11 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_11/stock_11_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK11;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_12/stock_12_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_12/stock_12_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_12/stock_12_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK12 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_12/stock_12_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK12;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_13/stock_13_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_13/stock_13_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_13/stock_13_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK13 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_13/stock_13_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK13;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_14/stock_14_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_14/stock_14_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_14/stock_14_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK14;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK14 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_14/stock_14_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK14;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_15/stock_15_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_15/stock_15_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_15/stock_15_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK15 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_15/stock_15_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK15;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_16/stock_16_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_16/stock_16_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_16/stock_16_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK16 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_16/stock_16_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK16;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_17/stock_17_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_17/stock_17_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_17/stock_17_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK17 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_17/stock_17_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK17;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_18/stock_18_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_18/stock_18_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_18/stock_18_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK18 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_18/stock_18_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK18;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_19/stock_19_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_19/stock_19_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_19/stock_19_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK19 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_19/stock_19_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK19;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_20/stock_20_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_20/stock_20_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_20/stock_20_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK20 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_20/stock_20_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK20;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_21/stock_21_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_21/stock_21_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_21/stock_21_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK21 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_21/stock_21_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK21;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_22/stock_22_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_22/stock_22_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_22/stock_22_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK22 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_22/stock_22_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK22;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_23/stock_23_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK23;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_23/stock_23_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_23/stock_23_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK23 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_23/stock_23_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK23;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_24/stock_24_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_24/stock_24_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_24/stock_24_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK24 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_24/stock_24_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK24;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_25/stock_25_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_25/stock_25_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_25/stock_25_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK25 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_25/stock_25_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK25;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_26/stock_26_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_26/stock_26_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_26/stock_26_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK26 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_26/stock_26_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK26;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_27/stock_27_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_27/stock_27_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_27/stock_27_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK27 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_27/stock_27_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK27;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_28/stock_28_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_28/stock_28_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_28/stock_28_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK28 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_28/stock_28_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK28;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_29/stock_29_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_29/stock_29_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_29/stock_29_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK29 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_29/stock_29_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK29;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_30/stock_30_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_30/stock_30_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_30/stock_30_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK30 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_30/stock_30_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK30;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/stock_31_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/stock_31_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK31;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/stock_31_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK31 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_31/stock_31_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK31;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/stock_32_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/stock_32_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/stock_32_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK32 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_32/stock_32_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK32;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/stock_33_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/stock_33_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/stock_33_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK33 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_33/stock_33_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK33;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/stock_34_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/stock_34_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/stock_34_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK34 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_34/stock_34_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK34;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/stock_35_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/stock_35_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/stock_35_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK35 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_35/stock_35_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK35;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/stock_36_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/stock_36_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/stock_36_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK36 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_36/stock_36_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK36;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/stock_37_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/stock_37_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/stock_37_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK37 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_37/stock_37_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK37;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/stock_38_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/stock_38_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/stock_38_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK38 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_38/stock_38_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK38;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_39/stock_39_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_39/stock_39_2.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK39;
COMMIT WORK;
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_39/stock_39_3.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK39;
COMMIT WORK;
```

```
CONNECT RESET;
CONNECT TO tpcc IN SHARE MODE;
ALTER TABLE STOCK39 ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM /flats/ua_39/stock_39_4.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 43500000 INSERT INTO STOCK39;
COMMIT WORK;
CONNECT RESET;
```

LOAD_WAREHOUSE_ALL.ddl

```
CONNECT TO tpcc IN SHARE MODE;
IMPORT FROM /flats/wdihn/warehouse_1.dat OF DEL MODIFIED BY COLDEL| KEEPBLANKS
COMPOUND=50 COMMITCOUNT 1000 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;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.CUSTOMER39 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;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.ORDER_LINE39 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;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.STOCK29 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.STOCK30 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.STOCK31 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.STOCK32 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.STOCK33 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.STOCK34 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.STOCK35 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.STOCK36 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.STOCK37 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.STOCK38 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to tpcc in share mode;
RUNSTATS ON TABLE TPCC.STOCK39 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 /flats/ord +5n +6n +7n /flats/ord2/orders_1.dat > /flats/ord2/orders_1.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_1.dat.sorted /flats/ord2/orders_1.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_1.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_1.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_2.dat > /flats/ord2/orders_2.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_2.dat.sorted /flats/ord2/orders_2.dat
else
  echo "**** WARNING ****"
```

```
 echo "Sort of /flats/ord2/orders_2.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_2.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_3.dat > /flats/ord2/orders_3.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_3.dat.sorted /flats/ord2/orders_3.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_3.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_3.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_4.dat > /flats/ord2/orders_4.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_4.dat.sorted /flats/ord2/orders_4.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_4.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_4.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_5.dat > /flats/ord2/orders_5.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_5.dat.sorted /flats/ord2/orders_5.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_5.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_5.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_6.dat > /flats/ord2/orders_6.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_6.dat.sorted /flats/ord2/orders_6.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_6.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_6.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_7.dat > /flats/ord2/orders_7.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_7.dat.sorted /flats/ord2/orders_7.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_7.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_7.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_8.dat > /flats/ord2/orders_8.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_8.dat.sorted /flats/ord2/orders_8.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_8.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_8.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_9.dat > /flats/ord2/orders_9.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_9.dat.sorted /flats/ord2/orders_9.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_9.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_9.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_10.dat > /flats/ord2/orders_10.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_10.dat.sorted /flats/ord2/orders_10.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_10.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_10.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_11.dat > /flats/ord2/orders_11.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_11.dat.sorted /flats/ord2/orders_11.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_11.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_11.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_12.dat > /flats/ord2/orders_12.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_12.dat.sorted /flats/ord2/orders_12.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_12.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_12.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_13.dat > /flats/ord2/orders_13.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_13.dat.sorted /flats/ord2/orders_13.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_13.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_13.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_14.dat > /flats/ord2/orders_14.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_14.dat.sorted /flats/ord2/orders_14.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_14.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_14.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_15.dat > /flats/ord2/orders_15.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_15.dat.sorted /flats/ord2/orders_15.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_15.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_15.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_16.dat > /flats/ord2/orders_16.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_16.dat.sorted /flats/ord2/orders_16.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_16.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_16.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_17.dat > /flats/ord2/orders_17.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_17.dat.sorted /flats/ord2/orders_17.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_17.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_17.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_18.dat > /flats/ord2/orders_18.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_18.dat.sorted /flats/ord2/orders_18.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_18.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_18.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_19.dat > /flats/ord2/orders_19.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_19.dat.sorted /flats/ord2/orders_19.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_19.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_19.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_20.dat > /flats/ord2/orders_20.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_20.dat.sorted /flats/ord2/orders_20.dat
else
 echo "**** WARNING ****"
```

```
 echo "Sort of /flats/ord2/orders_20.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_20.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_21.dat > /flats/ord2/orders_21.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_21.dat.sorted /flats/ord2/orders_21.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_21.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_21.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_22.dat > /flats/ord2/orders_22.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_22.dat.sorted /flats/ord2/orders_22.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_22.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_22.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_23.dat > /flats/ord2/orders_23.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_23.dat.sorted /flats/ord2/orders_23.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_23.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_23.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_24.dat > /flats/ord2/orders_24.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_24.dat.sorted /flats/ord2/orders_24.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_24.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_24.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_25.dat > /flats/ord2/orders_25.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_25.dat.sorted /flats/ord2/orders_25.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_25.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_25.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_26.dat > /flats/ord2/orders_26.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_26.dat.sorted /flats/ord2/orders_26.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_26.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_26.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_27.dat > /flats/ord2/orders_27.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_27.dat.sorted /flats/ord2/orders_27.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_27.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_27.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_28.dat > /flats/ord2/orders_28.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_28.dat.sorted /flats/ord2/orders_28.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_28.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_28.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_29.dat > /flats/ord2/orders_29.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_29.dat.sorted /flats/ord2/orders_29.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_29.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_29.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_30.dat > /flats/ord2/orders_30.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_30.dat.sorted /flats/ord2/orders_30.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_30.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_30.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_31.dat > /flats/ord2/orders_31.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_31.dat.sorted /flats/ord2/orders_31.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_31.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_31.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_32.dat > /flats/ord2/orders_32.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_32.dat.sorted /flats/ord2/orders_32.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_32.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_32.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_33.dat > /flats/ord2/orders_33.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_33.dat.sorted /flats/ord2/orders_33.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_33.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_33.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_34.dat > /flats/ord2/orders_34.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_34.dat.sorted /flats/ord2/orders_34.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_34.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_34.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_35.dat > /flats/ord2/orders_35.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_35.dat.sorted /flats/ord2/orders_35.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_35.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_35.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_36.dat > /flats/ord2/orders_36.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_36.dat.sorted /flats/ord2/orders_36.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_36.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_36.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_37.dat > /flats/ord2/orders_37.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_37.dat.sorted /flats/ord2/orders_37.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_37.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_37.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_38.dat > /flats/ord2/orders_38.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_38.dat.sorted /flats/ord2/orders_38.dat
else
 echo "**** WARNING ****"
```

```
  echo "Sort of /flats/ord2/orders_38.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_38.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_39.dat > /flats/ord2/orders_39.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_39.dat.sorted /flats/ord2/orders_39.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_39.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_39.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_40.dat > /flats/ord2/orders_40.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_40.dat.sorted /flats/ord2/orders_40.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_40.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_40.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_41.dat > /flats/ord2/orders_41.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_41.dat.sorted /flats/ord2/orders_41.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_41.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_41.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_42.dat > /flats/ord2/orders_42.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_42.dat.sorted /flats/ord2/orders_42.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_42.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_42.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_43.dat > /flats/ord2/orders_43.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_43.dat.sorted /flats/ord2/orders_43.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_43.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_43.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_44.dat > /flats/ord2/orders_44.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_44.dat.sorted /flats/ord2/orders_44.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_44.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_44.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_45.dat > /flats/ord2/orders_45.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_45.dat.sorted /flats/ord2/orders_45.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_45.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_45.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_46.dat > /flats/ord2/orders_46.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_46.dat.sorted /flats/ord2/orders_46.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_46.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_46.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_47.dat > /flats/ord2/orders_47.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_47.dat.sorted /flats/ord2/orders_47.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_47.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_47.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_48.dat > /flats/ord2/orders_48.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_48.dat.sorted /flats/ord2/orders_48.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_48.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_48.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_49.dat > /flats/ord2/orders_49.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_49.dat.sorted /flats/ord2/orders_49.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_49.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_49.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_50.dat > /flats/ord2/orders_50.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_50.dat.sorted /flats/ord2/orders_50.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_50.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_50.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_51.dat > /flats/ord2/orders_51.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_51.dat.sorted /flats/ord2/orders_51.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_51.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_51.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_52.dat > /flats/ord2/orders_52.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_52.dat.sorted /flats/ord2/orders_52.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_52.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_52.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_53.dat > /flats/ord2/orders_53.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_53.dat.sorted /flats/ord2/orders_53.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_53.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_53.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_54.dat > /flats/ord2/orders_54.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_54.dat.sorted /flats/ord2/orders_54.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_54.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_54.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_55.dat > /flats/ord2/orders_55.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_55.dat.sorted /flats/ord2/orders_55.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_55.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_55.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_56.dat > /flats/ord2/orders_56.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_56.dat.sorted /flats/ord2/orders_56.dat
else
  echo "**** WARNING ****"
```

```
 echo "Sort of /flats/ord2/orders_56.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_56.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_57.dat > /flats/ord2/orders_57.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_57.dat.sorted /flats/ord2/orders_57.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_57.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_57.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_58.dat > /flats/ord2/orders_58.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_58.dat.sorted /flats/ord2/orders_58.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_58.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_58.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_59.dat > /flats/ord2/orders_59.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_59.dat.sorted /flats/ord2/orders_59.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_59.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_59.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_60.dat > /flats/ord2/orders_60.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_60.dat.sorted /flats/ord2/orders_60.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_60.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_60.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_61.dat > /flats/ord2/orders_61.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_61.dat.sorted /flats/ord2/orders_61.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_61.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_61.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_62.dat > /flats/ord2/orders_62.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_62.dat.sorted /flats/ord2/orders_62.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_62.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_62.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_63.dat > /flats/ord2/orders_63.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_63.dat.sorted /flats/ord2/orders_63.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_63.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_63.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_64.dat > /flats/ord2/orders_64.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_64.dat.sorted /flats/ord2/orders_64.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_64.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_64.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_65.dat > /flats/ord2/orders_65.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_65.dat.sorted /flats/ord2/orders_65.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_65.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_65.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_66.dat > /flats/ord2/orders_66.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_66.dat.sorted /flats/ord2/orders_66.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_66.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_66.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_67.dat > /flats/ord2/orders_67.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_67.dat.sorted /flats/ord2/orders_67.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_67.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_67.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_68.dat > /flats/ord2/orders_68.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_68.dat.sorted /flats/ord2/orders_68.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_68.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_68.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_69.dat > /flats/ord2/orders_69.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_69.dat.sorted /flats/ord2/orders_69.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_69.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_69.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_70.dat > /flats/ord2/orders_70.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_70.dat.sorted /flats/ord2/orders_70.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_70.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_70.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_71.dat > /flats/ord2/orders_71.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_71.dat.sorted /flats/ord2/orders_71.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_71.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_71.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_72.dat > /flats/ord2/orders_72.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_72.dat.sorted /flats/ord2/orders_72.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_72.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_72.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_73.dat > /flats/ord2/orders_73.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_73.dat.sorted /flats/ord2/orders_73.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_73.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_73.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_74.dat > /flats/ord2/orders_74.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_74.dat.sorted /flats/ord2/orders_74.dat
else
 echo "**** WARNING ****"
```

```
 echo "Sort of /flats/ord2/orders_74.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_74.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_75.dat > /flats/ord2/orders_75.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_75.dat.sorted /flats/ord2/orders_75.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_75.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_75.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_76.dat > /flats/ord2/orders_76.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_76.dat.sorted /flats/ord2/orders_76.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_76.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_76.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_77.dat > /flats/ord2/orders_77.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_77.dat.sorted /flats/ord2/orders_77.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_77.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_77.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_78.dat > /flats/ord2/orders_78.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_78.dat.sorted /flats/ord2/orders_78.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_78.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_78.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_79.dat > /flats/ord2/orders_79.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_79.dat.sorted /flats/ord2/orders_79.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_79.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_79.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_80.dat > /flats/ord2/orders_80.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_80.dat.sorted /flats/ord2/orders_80.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_80.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_80.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_81.dat > /flats/ord2/orders_81.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_81.dat.sorted /flats/ord2/orders_81.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_81.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_81.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_82.dat > /flats/ord2/orders_82.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_82.dat.sorted /flats/ord2/orders_82.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_82.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_82.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_83.dat > /flats/ord2/orders_83.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_83.dat.sorted /flats/ord2/orders_83.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_83.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_83.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_84.dat > /flats/ord2/orders_84.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_84.dat.sorted /flats/ord2/orders_84.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_84.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_84.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_85.dat > /flats/ord2/orders_85.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_85.dat.sorted /flats/ord2/orders_85.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_85.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_85.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_86.dat > /flats/ord2/orders_86.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_86.dat.sorted /flats/ord2/orders_86.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_86.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_86.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_87.dat > /flats/ord2/orders_87.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_87.dat.sorted /flats/ord2/orders_87.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_87.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_87.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_88.dat > /flats/ord2/orders_88.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_88.dat.sorted /flats/ord2/orders_88.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_88.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_88.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_89.dat > /flats/ord2/orders_89.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_89.dat.sorted /flats/ord2/orders_89.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_89.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_89.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_90.dat > /flats/ord2/orders_90.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_90.dat.sorted /flats/ord2/orders_90.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_90.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_90.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_91.dat > /flats/ord2/orders_91.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_91.dat.sorted /flats/ord2/orders_91.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_91.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_91.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_92.dat > /flats/ord2/orders_92.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_92.dat.sorted /flats/ord2/orders_92.dat
else
 echo "**** WARNING ****"
```

```
 echo "Sort of /flats/ord2/orders_92.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_92.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_93.dat > /flats/ord2/orders_93.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_93.dat.sorted /flats/ord2/orders_93.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_93.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_93.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_94.dat > /flats/ord2/orders_94.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_94.dat.sorted /flats/ord2/orders_94.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_94.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_94.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_95.dat > /flats/ord2/orders_95.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_95.dat.sorted /flats/ord2/orders_95.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_95.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_95.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_96.dat > /flats/ord2/orders_96.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_96.dat.sorted /flats/ord2/orders_96.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_96.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_96.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_97.dat > /flats/ord2/orders_97.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_97.dat.sorted /flats/ord2/orders_97.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_97.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_97.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_98.dat > /flats/ord2/orders_98.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_98.dat.sorted /flats/ord2/orders_98.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_98.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_98.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_99.dat > /flats/ord2/orders_99.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_99.dat.sorted /flats/ord2/orders_99.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_99.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_99.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_100.dat > /flats/ord2/orders_100.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_100.dat.sorted /flats/ord2/orders_100.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_100.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_100.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_101.dat > /flats/ord2/orders_101.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_101.dat.sorted /flats/ord2/orders_101.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_101.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_101.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_102.dat > /flats/ord2/orders_102.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_102.dat.sorted /flats/ord2/orders_102.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_102.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_102.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_103.dat > /flats/ord2/orders_103.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_103.dat.sorted /flats/ord2/orders_103.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_103.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_103.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_104.dat > /flats/ord2/orders_104.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_104.dat.sorted /flats/ord2/orders_104.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_104.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_104.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_105.dat > /flats/ord2/orders_105.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_105.dat.sorted /flats/ord2/orders_105.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_105.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_105.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_106.dat > /flats/ord2/orders_106.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_106.dat.sorted /flats/ord2/orders_106.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_106.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_106.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_107.dat > /flats/ord2/orders_107.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_107.dat.sorted /flats/ord2/orders_107.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_107.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_107.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_108.dat > /flats/ord2/orders_108.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_108.dat.sorted /flats/ord2/orders_108.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_108.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_108.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_109.dat > /flats/ord2/orders_109.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_109.dat.sorted /flats/ord2/orders_109.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_109.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_109.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_110.dat > /flats/ord2/orders_110.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_110.dat.sorted /flats/ord2/orders_110.dat
else
 echo "**** WARNING ****"
```

```
 echo "Sort of /flats/ord2/orders_110.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_110.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_111.dat > /flats/ord2/orders_111.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_111.dat.sorted /flats/ord2/orders_111.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_111.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_111.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_112.dat > /flats/ord2/orders_112.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_112.dat.sorted /flats/ord2/orders_112.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_112.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_112.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_113.dat > /flats/ord2/orders_113.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_113.dat.sorted /flats/ord2/orders_113.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_113.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_113.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_114.dat > /flats/ord2/orders_114.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_114.dat.sorted /flats/ord2/orders_114.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_114.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_114.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_115.dat > /flats/ord2/orders_115.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_115.dat.sorted /flats/ord2/orders_115.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_115.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_115.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_116.dat > /flats/ord2/orders_116.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_116.dat.sorted /flats/ord2/orders_116.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_116.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_116.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_117.dat > /flats/ord2/orders_117.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_117.dat.sorted /flats/ord2/orders_117.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_117.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_117.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_118.dat > /flats/ord2/orders_118.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_118.dat.sorted /flats/ord2/orders_118.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_118.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_118.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_119.dat > /flats/ord2/orders_119.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_119.dat.sorted /flats/ord2/orders_119.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_119.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_119.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_120.dat > /flats/ord2/orders_120.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_120.dat.sorted /flats/ord2/orders_120.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_120.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_120.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_121.dat > /flats/ord2/orders_121.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_121.dat.sorted /flats/ord2/orders_121.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_121.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_121.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_122.dat > /flats/ord2/orders_122.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_122.dat.sorted /flats/ord2/orders_122.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_122.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_122.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_123.dat > /flats/ord2/orders_123.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_123.dat.sorted /flats/ord2/orders_123.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_123.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_123.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_124.dat > /flats/ord2/orders_124.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_124.dat.sorted /flats/ord2/orders_124.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_124.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_124.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_125.dat > /flats/ord2/orders_125.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_125.dat.sorted /flats/ord2/orders_125.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_125.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_125.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_126.dat > /flats/ord2/orders_126.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_126.dat.sorted /flats/ord2/orders_126.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_126.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_126.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_127.dat > /flats/ord2/orders_127.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_127.dat.sorted /flats/ord2/orders_127.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_127.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_127.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_128.dat > /flats/ord2/orders_128.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_128.dat.sorted /flats/ord2/orders_128.dat
else
 echo "**** WARNING ****"
```

```
 echo "Sort of /flats/ord2/orders_128.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_128.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_129.dat > /flats/ord2/orders_129.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_129.dat.sorted /flats/ord2/orders_129.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_129.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_129.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_130.dat > /flats/ord2/orders_130.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_130.dat.sorted /flats/ord2/orders_130.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_130.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_130.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_131.dat > /flats/ord2/orders_131.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_131.dat.sorted /flats/ord2/orders_131.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_131.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_131.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_132.dat > /flats/ord2/orders_132.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_132.dat.sorted /flats/ord2/orders_132.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_132.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_132.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_133.dat > /flats/ord2/orders_133.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_133.dat.sorted /flats/ord2/orders_133.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_133.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_133.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_134.dat > /flats/ord2/orders_134.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_134.dat.sorted /flats/ord2/orders_134.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_134.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_134.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_135.dat > /flats/ord2/orders_135.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_135.dat.sorted /flats/ord2/orders_135.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_135.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_135.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_136.dat > /flats/ord2/orders_136.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_136.dat.sorted /flats/ord2/orders_136.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_136.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_136.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_137.dat > /flats/ord2/orders_137.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_137.dat.sorted /flats/ord2/orders_137.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_137.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_137.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_138.dat > /flats/ord2/orders_138.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_138.dat.sorted /flats/ord2/orders_138.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_138.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_138.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_139.dat > /flats/ord2/orders_139.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_139.dat.sorted /flats/ord2/orders_139.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_139.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_139.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_140.dat > /flats/ord2/orders_140.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_140.dat.sorted /flats/ord2/orders_140.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_140.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_140.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_141.dat > /flats/ord2/orders_141.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_141.dat.sorted /flats/ord2/orders_141.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_141.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_141.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_142.dat > /flats/ord2/orders_142.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_142.dat.sorted /flats/ord2/orders_142.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_142.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_142.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_143.dat > /flats/ord2/orders_143.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_143.dat.sorted /flats/ord2/orders_143.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_143.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_143.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_144.dat > /flats/ord2/orders_144.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_144.dat.sorted /flats/ord2/orders_144.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_144.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_144.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_145.dat > /flats/ord2/orders_145.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_145.dat.sorted /flats/ord2/orders_145.dat
else
 echo "**** WARNING ****"
 echo "Sort of /flats/ord2/orders_145.dat FAILED."
 echo "Please ensure that the source file (/flats/ord2/orders_145.dat)"
 echo "exists and that sufficient free space is available"
 echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_146.dat > /flats/ord2/orders_146.dat.sorted
if [ $? = 0 ]
then
 mv /flats/ord2/orders_146.dat.sorted /flats/ord2/orders_146.dat
else
 echo "**** WARNING ****"
```

```
  echo "Sort of /flats/ord2/orders_146.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_146.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_147.dat > /flats/ord2/orders_147.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_147.dat.sorted /flats/ord2/orders_147.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_147.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_147.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_148.dat > /flats/ord2/orders_148.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_148.dat.sorted /flats/ord2/orders_148.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_148.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_148.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_149.dat > /flats/ord2/orders_149.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_149.dat.sorted /flats/ord2/orders_149.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_149.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_149.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_150.dat > /flats/ord2/orders_150.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_150.dat.sorted /flats/ord2/orders_150.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_150.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_150.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_151.dat > /flats/ord2/orders_151.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_151.dat.sorted /flats/ord2/orders_151.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_151.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_151.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_152.dat > /flats/ord2/orders_152.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_152.dat.sorted /flats/ord2/orders_152.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_152.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_152.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_153.dat > /flats/ord2/orders_153.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_153.dat.sorted /flats/ord2/orders_153.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_153.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_153.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_154.dat > /flats/ord2/orders_154.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_154.dat.sorted /flats/ord2/orders_154.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_154.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_154.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_155.dat > /flats/ord2/orders_155.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_155.dat.sorted /flats/ord2/orders_155.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_155.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_155.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
sort -t'|' -T /flats/ord +5n +6n +7n /flats/ord2/orders_156.dat > /flats/ord2/orders_156.dat.sorted
if [ $? = 0 ]
then
  mv /flats/ord2/orders_156.dat.sorted /flats/ord2/orders_156.dat
else
  echo "**** WARNING ****"
  echo "Sort of /flats/ord2/orders_156.dat FAILED."
  echo "Please ensure that the source file (/flats/ord2/orders_156.dat)"
  echo "exists and that sufficient free space is available"
  echo "in SORTTMP (/flats/ord)."
fi
```

alter_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.
-----------------------------------------------------------------------------

-- Alter Size of Bufferpools
connect to tpcc;
alter bufferpool IBMDEFAULT    DEFERRED SIZE     1000;
alter bufferpool IBMDEFAULT8K  DEFERRED SIZE     1000;
alter bufferpool IBMDEFAULT16K DEFERRED SIZE     1000;
alter bufferpool WAR           DEFERRED SIZE     2000;
alter bufferpool DIS           DEFERRED SIZE    20000;
alter bufferpool ITM           DEFERRED SIZE     2000;
alter bufferpool HST           DEFERRED SIZE     1500;
alter bufferpool NEW           DEFERRED SIZE   500000;
alter bufferpool NEW_I         DEFERRED SIZE  1300000;
alter bufferpool ORD           DEFERRED SIZE  1000000;
alter bufferpool ORD_I         DEFERRED SIZE  3600000;
alter bufferpool OLN           DEFERRED SIZE  2700000;
alter bufferpool CST           DEFERRED SIZE  5265000;
alter bufferpool CST_I         DEFERRED SIZE  1850000;
alter bufferpool STK           DEFERRED SIZE 88675000;
connect reset;
terminate;
```

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_ORDERLINE_29 bufferpool OLN;
alter tablespace TS_ORDERLINE_30 bufferpool OLN;
alter tablespace TS_ORDERLINE_31 bufferpool OLN;
alter tablespace TS_ORDERLINE_32 bufferpool OLN;
alter tablespace TS_ORDERLINE_33 bufferpool OLN;
alter tablespace TS_ORDERLINE_34 bufferpool OLN;
alter tablespace TS_ORDERLINE_35 bufferpool OLN;
alter tablespace TS_ORDERLINE_36 bufferpool OLN;
alter tablespace TS_ORDERLINE_37 bufferpool OLN;
alter tablespace TS_ORDERLINE_38 bufferpool OLN;
alter tablespace TS_ORDERLINE_39 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 TS_CUSTOMER_29 bufferpool CST;
alter tablespace TS_CUSTOMER_30 bufferpool CST;
alter tablespace TS_CUSTOMER_31 bufferpool CST;
alter tablespace TS_CUSTOMER_32 bufferpool CST;
alter tablespace TS_CUSTOMER_33 bufferpool CST;
alter tablespace TS_CUSTOMER_34 bufferpool CST;
alter tablespace TS_CUSTOMER_35 bufferpool CST;
alter tablespace TS_CUSTOMER_36 bufferpool CST;
alter tablespace TS_CUSTOMER_37 bufferpool CST;
alter tablespace TS_CUSTOMER_38 bufferpool CST;
alter tablespace TS_CUSTOMER_39 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 IS_CUSTOMER_29 bufferpool CST_I;
alter tablespace IS_CUSTOMER_30 bufferpool CST_I;
alter tablespace IS_CUSTOMER_31 bufferpool CST_I;
alter tablespace IS_CUSTOMER_32 bufferpool CST_I;
alter tablespace IS_CUSTOMER_33 bufferpool CST_I;
alter tablespace IS_CUSTOMER_34 bufferpool CST_I;
alter tablespace IS_CUSTOMER_35 bufferpool CST_I;
alter tablespace IS_CUSTOMER_36 bufferpool CST_I;
alter tablespace IS_CUSTOMER_37 bufferpool CST_I;
alter tablespace IS_CUSTOMER_38 bufferpool CST_I;
alter tablespace IS_CUSTOMER_39 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;
alter tablespace TS_STOCK_29 bufferpool STK;
alter tablespace TS_STOCK_30 bufferpool STK;
alter tablespace TS_STOCK_31 bufferpool STK;
alter tablespace TS_STOCK_32 bufferpool STK;
alter tablespace TS_STOCK_33 bufferpool STK;
alter tablespace TS_STOCK_34 bufferpool STK;
alter tablespace TS_STOCK_35 bufferpool STK;
alter tablespace TS_STOCK_36 bufferpool STK;
alter tablespace TS_STOCK_37 bufferpool STK;
alter tablespace TS_STOCK_38 bufferpool STK;
alter tablespace TS_STOCK_39 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;

-- now creating TS for ts_ware_01 of D1
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/rD1F01V3WARE' 128,
                                device '/dev/rD1F02V1WARE' 128,
                                device '/dev/rD1F02V2WARE' 128,
                                device '/dev/rD1F02V3WARE' 128,
                                device '/dev/rD1F03V1WARE' 128,
                                device '/dev/rD1F03V2WARE' 128,
                                device '/dev/rD1F03V3WARE' 128,
                                device '/dev/rD1F04V1WARE' 128,
                                device '/dev/rD1F04V2WARE' 128,
                                device '/dev/rD1F04V3WARE' 128,
                                device '/dev/rD1F05V1WARE' 128,
                                device '/dev/rD1F05V2WARE' 128,
                                device '/dev/rD1F05V3WARE' 128,
                                device '/dev/rD1F06V1WARE' 128,
                                device '/dev/rD1F06V2WARE' 128,
                                device '/dev/rD1F06V3WARE' 128,
                                device '/dev/rD1F07V1WARE' 128,
                                device '/dev/rD1F07V2WARE' 128,
                                device '/dev/rD1F07V3WARE' 128,
                                device '/dev/rD1F08V1WARE' 128,
                                device '/dev/rD1F08V2WARE' 128,
                                device '/dev/rD1F08V3WARE' 128,
                                device '/dev/rD1F09V1WARE' 128,
                                device '/dev/rD1F09V2WARE' 128,
                                device '/dev/rD1F09V3WARE' 128,
                                device '/dev/rD1F10V1WARE' 128,
                                device '/dev/rD1F10V2WARE' 128,
                                device '/dev/rD1F10V3WARE' 128,
                                device '/dev/rD1F11V1WARE' 128,
                                device '/dev/rD1F11V2WARE' 128,
                                device '/dev/rD1F11V3WARE' 128,
                                device '/dev/rD1F12V1WARE' 128,
                                device '/dev/rD1F12V2WARE' 128,
                                device '/dev/rD1F12V3WARE' 128,
                                device '/dev/rD1F13V1WARE' 128,
                                device '/dev/rD1F13V2WARE' 128,
                                device '/dev/rD1F13V3WARE' 128
                                )
                extentsize 32
                prefetchsize 512;
commit;


-- now creating TS for ts_dist_01 of D1
drop tablespace ts_dist_01;
create regular tablespace ts_dist_01 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F01V1DIST' 1152,
                                device '/dev/rD1F01V2DIST' 1152,
                                device '/dev/rD1F01V3DIST' 1152,
                                device '/dev/rD1F02V1DIST' 1152,
                                device '/dev/rD1F02V2DIST' 1152,
                                device '/dev/rD1F02V3DIST' 1152,
                                device '/dev/rD1F03V1DIST' 1152,
                                device '/dev/rD1F03V2DIST' 1152,
                                device '/dev/rD1F03V3DIST' 1152,
                                device '/dev/rD1F04V1DIST' 1152,
                                device '/dev/rD1F04V2DIST' 1152,
                                device '/dev/rD1F04V3DIST' 1152,
                                device '/dev/rD1F05V1DIST' 1152,
                                device '/dev/rD1F05V2DIST' 1152,
                                device '/dev/rD1F05V3DIST' 1152,
                                device '/dev/rD1F06V1DIST' 1152,
                                device '/dev/rD1F06V2DIST' 1152,
                                device '/dev/rD1F06V3DIST' 1152,
                                device '/dev/rD1F07V1DIST' 1152,
                                device '/dev/rD1F07V2DIST' 1152,
                                device '/dev/rD1F07V3DIST' 1152,
                                device '/dev/rD1F08V1DIST' 1152,
                                device '/dev/rD1F08V2DIST' 1152,
                                device '/dev/rD1F08V3DIST' 1152,
                                device '/dev/rD1F09V1DIST' 1152,
                                device '/dev/rD1F09V2DIST' 1152,
                                device '/dev/rD1F09V3DIST' 1152,
                                device '/dev/rD1F10V1DIST' 1152,
                                device '/dev/rD1F10V2DIST' 1152,
                                device '/dev/rD1F10V3DIST' 1152,
                                device '/dev/rD1F11V1DIST' 1152,
                                device '/dev/rD1F11V2DIST' 1152,
                                device '/dev/rD1F11V3DIST' 1152,
                                device '/dev/rD1F12V1DIST' 1152,
                                device '/dev/rD1F12V2DIST' 1152,
                                device '/dev/rD1F12V3DIST' 1152,
                                device '/dev/rD1F13V1DIST' 1152,
                                device '/dev/rD1F13V2DIST' 1152,
                                device '/dev/rD1F13V3DIST' 1152
                                )
                extentsize 64
                prefetchsize 512;
commit;


-- now creating TS for ts_item_01 of D1
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/rD1F01V3ITEM' 128,
                                device '/dev/rD1F02V1ITEM' 128,
                                device '/dev/rD1F02V2ITEM' 128,
                                device '/dev/rD1F02V3ITEM' 128,
                                device '/dev/rD1F03V1ITEM' 128,
                                device '/dev/rD1F03V2ITEM' 128,
                                device '/dev/rD1F03V3ITEM' 128,
                                device '/dev/rD1F04V1ITEM' 128,
                                device '/dev/rD1F04V2ITEM' 128,
                                device '/dev/rD1F04V3ITEM' 128,
                                device '/dev/rD1F05V1ITEM' 128,
                                device '/dev/rD1F05V2ITEM' 128,
                                device '/dev/rD1F05V3ITEM' 128,
                                device '/dev/rD1F06V1ITEM' 128,
                                device '/dev/rD1F06V2ITEM' 128,
                                device '/dev/rD1F06V3ITEM' 128,
                                device '/dev/rD1F07V1ITEM' 128,
                                device '/dev/rD1F07V2ITEM' 128,
                                device '/dev/rD1F07V3ITEM' 128,
                                device '/dev/rD1F08V1ITEM' 128,
                                device '/dev/rD1F08V2ITEM' 128,
                                device '/dev/rD1F08V3ITEM' 128,
                                device '/dev/rD1F09V1ITEM' 128,
                                device '/dev/rD1F09V2ITEM' 128,
                                device '/dev/rD1F09V3ITEM' 128,
                                device '/dev/rD1F10V1ITEM' 128,
                                device '/dev/rD1F10V2ITEM' 128,
                                device '/dev/rD1F10V3ITEM' 128,
                                device '/dev/rD1F11V1ITEM' 128,
                                device '/dev/rD1F11V2ITEM' 128,
                                device '/dev/rD1F11V3ITEM' 128,
                                device '/dev/rD1F12V1ITEM' 128,
                                device '/dev/rD1F12V2ITEM' 128,
                                device '/dev/rD1F12V3ITEM' 128,
                                device '/dev/rD1F13V1ITEM' 128,
                                device '/dev/rD1F13V2ITEM' 128,
                                device '/dev/rD1F13V3ITEM' 128
                                )
                extentsize 32
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_stock_01 of D1

drop tablespace ts_stock_01;
create regular tablespace ts_stock_01 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F01V1STK1' 15386112
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_02 of D1

drop tablespace ts_stock_02;
create regular tablespace ts_stock_02 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F01V2STK2' 15386112
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_03 of D1

drop tablespace ts_stock_03;
create regular tablespace ts_stock_03 pagesize 4K
                managed by database
```

```
                using
                (
                        device '/dev/rD1F01V3STK3' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_04 of D1

drop tablespace ts_stock_04;
create regular tablespace ts_stock_04 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F02V1STK4' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_05 of D1

drop tablespace ts_stock_05;
create regular tablespace ts_stock_05 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F02V2STK5' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_06 of D1

drop tablespace ts_stock_06;
create regular tablespace ts_stock_06 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F02V3STK6' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_07 of D1

drop tablespace ts_stock_07;
create regular tablespace ts_stock_07 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F03V1STK7' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_08 of D1

drop tablespace ts_stock_08;
create regular tablespace ts_stock_08 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F03V2STK8' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_09 of D1

drop tablespace ts_stock_09;
create regular tablespace ts_stock_09 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F03V3STK9' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_10 of D1

drop tablespace ts_stock_10;
create regular tablespace ts_stock_10 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F04V1STK10' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_11 of D1

drop tablespace ts_stock_11;
create regular tablespace ts_stock_11 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F04V2STK11' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_12 of D1

drop tablespace ts_stock_12;
create regular tablespace ts_stock_12 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F04V3STK12' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_13 of D1

drop tablespace ts_stock_13;
create regular tablespace ts_stock_13 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F05V1STK13' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_14 of D1

drop tablespace ts_stock_14;
create regular tablespace ts_stock_14 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F05V2STK14' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_15 of D1

drop tablespace ts_stock_15;
create regular tablespace ts_stock_15 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F05V3STK15' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_16 of D1

drop tablespace ts_stock_16;
create regular tablespace ts_stock_16 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F06V1STK16' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_17 of D1

drop tablespace ts_stock_17;
create regular tablespace ts_stock_17 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F06V2STK17' 15386112
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_18 of D1

drop tablespace ts_stock_18;
create regular tablespace ts_stock_18 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F06V3STK18' 15386112
```

```
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_19 of D1

drop tablespace ts_stock_19;
create regular tablespace ts_stock_19 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F07V1STK19' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_20 of D1

drop tablespace ts_stock_20;
create regular tablespace ts_stock_20 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F07V2STK20' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_21 of D1

drop tablespace ts_stock_21;
create regular tablespace ts_stock_21 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F07V3STK21' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_22 of D1

drop tablespace ts_stock_22;
create regular tablespace ts_stock_22 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F08V1STK22' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_23 of D1

drop tablespace ts_stock_23;
create regular tablespace ts_stock_23 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F08V2STK23' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_24 of D1

drop tablespace ts_stock_24;
create regular tablespace ts_stock_24 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F08V3STK24' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_25 of D1

drop tablespace ts_stock_25;
create regular tablespace ts_stock_25 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F09V1STK25' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_26 of D1

drop tablespace ts_stock_26;
create regular tablespace ts_stock_26 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F09V2STK26' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_27 of D1

drop tablespace ts_stock_27;
create regular tablespace ts_stock_27 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F09V3STK27' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_28 of D1

drop tablespace ts_stock_28;
create regular tablespace ts_stock_28 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F10V1STK28' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_29 of D1

drop tablespace ts_stock_29;
create regular tablespace ts_stock_29 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F10V2STK29' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_30 of D1

drop tablespace ts_stock_30;
create regular tablespace ts_stock_30 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F10V3STK30' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_31 of D1

drop tablespace ts_stock_31;
create regular tablespace ts_stock_31 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F11V1STK31' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_32 of D1

drop tablespace ts_stock_32;
create regular tablespace ts_stock_32 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F11V2STK32' 15386112
                    )
            extentsize 256
            prefetchsize 512;
commit;

-- now creating TS for ts_stock_33 of D1

drop tablespace ts_stock_33;
create regular tablespace ts_stock_33 pagesize 4K
            managed by database
            using
            (
                    device '/dev/rD1F11V3STK33' 15386112
                    )
            extentsize 256
            prefetchsize 512;
```

```
commit;

-- now creating TS for ts_stock_34 of D1

drop tablespace ts_stock_34;
create regular tablespace ts_stock_34 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F12V1STK34' 15386112
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_35 of D1

drop tablespace ts_stock_35;
create regular tablespace ts_stock_35 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F12V2STK35' 15386112
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_36 of D1

drop tablespace ts_stock_36;
create regular tablespace ts_stock_36 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F12V3STK36' 15386112
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_37 of D1

drop tablespace ts_stock_37;
create regular tablespace ts_stock_37 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F13V1STK37' 15386112
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_38 of D1

drop tablespace ts_stock_38;
create regular tablespace ts_stock_38 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F13V2STK38' 15386112
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_stock_39 of D1

drop tablespace ts_stock_39;
create regular tablespace ts_stock_39 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F13V3STK39' 15386112
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_01 of D1

drop tablespace ts_customer_01;
create regular tablespace ts_customer_01 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F01V1CST1' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_02 of D1

drop tablespace ts_customer_02;
create regular tablespace ts_customer_02 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F01V2CST2' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_03 of D1

drop tablespace ts_customer_03;
create regular tablespace ts_customer_03 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F01V3CST3' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_04 of D1

drop tablespace ts_customer_04;
create regular tablespace ts_customer_04 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F02V1CST4' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_05 of D1

drop tablespace ts_customer_05;
create regular tablespace ts_customer_05 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F02V2CST5' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_06 of D1

drop tablespace ts_customer_06;
create regular tablespace ts_customer_06 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F02V3CST6' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_07 of D1

drop tablespace ts_customer_07;
create regular tablespace ts_customer_07 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F03V1CST7' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_08 of D1

drop tablespace ts_customer_08;
create regular tablespace ts_customer_08 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F03V2CST8' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_09 of D1

drop tablespace ts_customer_09;
create regular tablespace ts_customer_09 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F03V3CST9' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_10 of D1
```

```
drop tablespace ts_customer_10;
create regular tablespace ts_customer_10 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F04V1CST10' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_11 of D1

drop tablespace ts_customer_11;
create regular tablespace ts_customer_11 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F04V2CST11' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_12 of D1

drop tablespace ts_customer_12;
create regular tablespace ts_customer_12 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F04V3CST12' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_13 of D1

drop tablespace ts_customer_13;
create regular tablespace ts_customer_13 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F05V1CST13' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_14 of D1

drop tablespace ts_customer_14;
create regular tablespace ts_customer_14 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F05V2CST14' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_15 of D1

drop tablespace ts_customer_15;
create regular tablespace ts_customer_15 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F05V3CST15' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_16 of D1

drop tablespace ts_customer_16;
create regular tablespace ts_customer_16 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F06V1CST16' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_17 of D1

drop tablespace ts_customer_17;
create regular tablespace ts_customer_17 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F06V2CST17' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_18 of D1

drop tablespace ts_customer_18;
create regular tablespace ts_customer_18 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F06V3CST18' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_19 of D1

drop tablespace ts_customer_19;
create regular tablespace ts_customer_19 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F07V1CST19' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_20 of D1

drop tablespace ts_customer_20;
create regular tablespace ts_customer_20 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F07V2CST20' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_21 of D1

drop tablespace ts_customer_21;
create regular tablespace ts_customer_21 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F07V3CST21' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_22 of D1

drop tablespace ts_customer_22;
create regular tablespace ts_customer_22 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F08V1CST22' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_23 of D1

drop tablespace ts_customer_23;
create regular tablespace ts_customer_23 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F08V2CST23' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_24 of D1

drop tablespace ts_customer_24;
create regular tablespace ts_customer_24 pagesize 4K
                managed by database
                using
                (
                                device '/dev/rD1F08V3CST24' 10973952
                                )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_25 of D1

drop tablespace ts_customer_25;
create regular tablespace ts_customer_25 pagesize 4K
```

```
                managed by database
                using
                (
                        device '/dev/rD1F09V1CST25' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_26 of D1

drop tablespace ts_customer_26;
create regular tablespace ts_customer_26 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F09V2CST26' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_27 of D1

drop tablespace ts_customer_27;
create regular tablespace ts_customer_27 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F09V3CST27' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_28 of D1

drop tablespace ts_customer_28;
create regular tablespace ts_customer_28 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F10V1CST28' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_29 of D1

drop tablespace ts_customer_29;
create regular tablespace ts_customer_29 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F10V2CST29' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_30 of D1

drop tablespace ts_customer_30;
create regular tablespace ts_customer_30 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F10V3CST30' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_31 of D1

drop tablespace ts_customer_31;
create regular tablespace ts_customer_31 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F11V1CST31' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_32 of D1

drop tablespace ts_customer_32;
create regular tablespace ts_customer_32 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F11V2CST32' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_33 of D1

drop tablespace ts_customer_33;
create regular tablespace ts_customer_33 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F11V3CST33' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_34 of D1

drop tablespace ts_customer_34;
create regular tablespace ts_customer_34 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F12V1CST34' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_35 of D1

drop tablespace ts_customer_35;
create regular tablespace ts_customer_35 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F12V2CST35' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_36 of D1

drop tablespace ts_customer_36;
create regular tablespace ts_customer_36 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F12V3CST36' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_37 of D1

drop tablespace ts_customer_37;
create regular tablespace ts_customer_37 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F13V1CST37' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_38 of D1

drop tablespace ts_customer_38;
create regular tablespace ts_customer_38 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F13V2CST38' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for ts_customer_39 of D1

drop tablespace ts_customer_39;
create regular tablespace ts_customer_39 pagesize 4K
                managed by database
                using
                (
                        device '/dev/rD1F13V3CST39' 10973952
                        )
                extentsize 256
                prefetchsize 512;
commit;

-- now creating TS for is_customer_01 of D1

drop tablespace is_customer_01;
create regular tablespace is_customer_01 pagesize 8K
                managed by database
                using
                (
```

```
                    device '/dev/rD1F01V1CSTI1' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_02 of D1

drop tablespace is_customer_02;
create regular tablespace is_customer_02 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F01V2CSTI2' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_03 of D1

drop tablespace is_customer_03;
create regular tablespace is_customer_03 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F01V3CSTI3' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_04 of D1

drop tablespace is_customer_04;
create regular tablespace is_customer_04 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F02V1CSTI4' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_05 of D1

drop tablespace is_customer_05;
create regular tablespace is_customer_05 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F02V2CSTI5' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_06 of D1

drop tablespace is_customer_06;
create regular tablespace is_customer_06 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F02V3CSTI6' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_07 of D1

drop tablespace is_customer_07;
create regular tablespace is_customer_07 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F03V1CSTI7' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_08 of D1

drop tablespace is_customer_08;
create regular tablespace is_customer_08 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F03V2CSTI8' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_09 of D1

drop tablespace is_customer_09;
create regular tablespace is_customer_09 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F03V3CSTI9' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_10 of D1

drop tablespace is_customer_10;
create regular tablespace is_customer_10 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F04V1CSTI10' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_11 of D1

drop tablespace is_customer_11;
create regular tablespace is_customer_11 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F04V2CSTI11' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_12 of D1

drop tablespace is_customer_12;
create regular tablespace is_customer_12 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F04V3CSTI12' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_13 of D1

drop tablespace is_customer_13;
create regular tablespace is_customer_13 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F05V1CSTI13' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_14 of D1

drop tablespace is_customer_14;
create regular tablespace is_customer_14 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F05V2CSTI14' 386048
                    )
          extentsize 256
          prefetchsize 512
          bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_15 of D1

drop tablespace is_customer_15;
create regular tablespace is_customer_15 pagesize 8K
          managed by database
          using
          (
                    device '/dev/rD1F05V3CSTI15' 386048
                    )
```

```
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_16 of D1

drop tablespace is_customer_16;
create regular tablespace is_customer_16 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F06V1CSTI16' 386048
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_17 of D1

drop tablespace is_customer_17;
create regular tablespace is_customer_17 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F06V2CSTI17' 386048
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_18 of D1

drop tablespace is_customer_18;
create regular tablespace is_customer_18 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F06V3CSTI18' 386048
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_19 of D1

drop tablespace is_customer_19;
create regular tablespace is_customer_19 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F07V1CSTI19' 386048
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_20 of D1

drop tablespace is_customer_20;
create regular tablespace is_customer_20 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F07V2CSTI20' 386048
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_21 of D1

drop tablespace is_customer_21;
create regular tablespace is_customer_21 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F07V3CSTI21' 386048
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_22 of D1

drop tablespace is_customer_22;
create regular tablespace is_customer_22 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F08V1CSTI22' 386048
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_23 of D1

drop tablespace is_customer_23;
create regular tablespace is_customer_23 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F08V2CSTI23' 386048
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_24 of D1

drop tablespace is_customer_24;
create regular tablespace is_customer_24 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F08V3CSTI24' 386048
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_25 of D1

drop tablespace is_customer_25;
create regular tablespace is_customer_25 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F09V1CSTI25' 386048
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_26 of D1

drop tablespace is_customer_26;
create regular tablespace is_customer_26 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F09V2CSTI26' 386048
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_27 of D1

drop tablespace is_customer_27;
create regular tablespace is_customer_27 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F09V3CSTI27' 386048
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_28 of D1

drop tablespace is_customer_28;
create regular tablespace is_customer_28 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F10V1CSTI28' 386048
                               )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_29 of D1

drop tablespace is_customer_29;
create regular tablespace is_customer_29 pagesize 8K
                 managed by database
                 using
                (
                                device '/dev/rD1F10V2CSTI29' 386048
                               )
                extentsize 256
                prefetchsize 512
```

```
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_30 of D1

drop tablespace is_customer_30;
create regular tablespace is_customer_30 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F10V3CSTI30' 386048
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_31 of D1

drop tablespace is_customer_31;
create regular tablespace is_customer_31 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F11V1CSTI31' 386048
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_32 of D1

drop tablespace is_customer_32;
create regular tablespace is_customer_32 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F11V2CSTI32' 386048
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_33 of D1

drop tablespace is_customer_33;
create regular tablespace is_customer_33 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F11V3CSTI33' 386048
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_34 of D1

drop tablespace is_customer_34;
create regular tablespace is_customer_34 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F12V1CSTI34' 386048
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_35 of D1

drop tablespace is_customer_35;
create regular tablespace is_customer_35 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F12V2CSTI35' 386048
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_36 of D1

drop tablespace is_customer_36;
create regular tablespace is_customer_36 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F12V3CSTI36' 386048
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_37 of D1

drop tablespace is_customer_37;
create regular tablespace is_customer_37 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F13V1CSTI37' 386048
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_38 of D1

drop tablespace is_customer_38;
create regular tablespace is_customer_38 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F13V2CSTI38' 386048
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for is_customer_39 of D1

drop tablespace is_customer_39;
create regular tablespace is_customer_39 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F13V3CSTI39' 386048
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;


-- now creating TS for ts_history_01 of D1
drop tablespace ts_history_01;
create regular tablespace ts_history_01 pagesize 16K
                managed by database
                using
                (
                                device '/dev/rD1F01V1HIST' 321536,
                                device '/dev/rD1F01V2HIST' 321536,
                                device '/dev/rD1F01V3HIST' 321536,
                                device '/dev/rD1F02V1HIST' 321536,
                                device '/dev/rD1F02V2HIST' 321536,
                                device '/dev/rD1F02V3HIST' 321536,
                                device '/dev/rD1F03V1HIST' 321536,
                                device '/dev/rD1F03V2HIST' 321536,
                                device '/dev/rD1F03V3HIST' 321536,
                                device '/dev/rD1F04V1HIST' 321536,
                                device '/dev/rD1F04V2HIST' 321536,
                                device '/dev/rD1F04V3HIST' 321536,
                                device '/dev/rD1F05V1HIST' 321536,
                                device '/dev/rD1F05V2HIST' 321536,
                                device '/dev/rD1F05V3HIST' 321536,
                                device '/dev/rD1F06V1HIST' 321536,
                                device '/dev/rD1F06V2HIST' 321536,
                                device '/dev/rD1F06V3HIST' 321536,
                                device '/dev/rD1F07V1HIST' 321536,
                                device '/dev/rD1F07V2HIST' 321536,
                                device '/dev/rD1F07V3HIST' 321536,
                                device '/dev/rD1F08V1HIST' 321536,
                                device '/dev/rD1F08V2HIST' 321536,
                                device '/dev/rD1F08V3HIST' 321536,
                                device '/dev/rD1F09V1HIST' 321536,
                                device '/dev/rD1F09V2HIST' 321536,
                                device '/dev/rD1F09V3HIST' 321536,
                                device '/dev/rD1F10V1HIST' 321536,
                                device '/dev/rD1F10V2HIST' 321536,
                                device '/dev/rD1F10V3HIST' 321536,
                                device '/dev/rD1F11V1HIST' 321536,
                                device '/dev/rD1F11V2HIST' 321536,
                                device '/dev/rD1F11V3HIST' 321536,
                                device '/dev/rD1F12V1HIST' 321536,
                                device '/dev/rD1F12V2HIST' 321536,
                                device '/dev/rD1F12V3HIST' 321536,
                                device '/dev/rD1F13V1HIST' 321536,
                                device '/dev/rD1F13V2HIST' 321536,
                                device '/dev/rD1F13V3HIST' 321536
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp16K;
commit;


-- now creating TS for ts_order_01 of D1
drop tablespace ts_order_01;
```

```
create regular tablespace ts_order_01 pagesize 8K
                managed by database
                using
               (
                                device '/dev/rD1F01V1ORD' 339456,
                                device '/dev/rD1F01V2ORD' 339456,
                                device '/dev/rD1F01V3ORD' 339456,
                                device '/dev/rD1F02V1ORD' 339456,
                                device '/dev/rD1F02V2ORD' 339456,
                                device '/dev/rD1F02V3ORD' 339456,
                                device '/dev/rD1F03V1ORD' 339456,
                                device '/dev/rD1F03V2ORD' 339456,
                                device '/dev/rD1F03V3ORD' 339456,
                                device '/dev/rD1F04V1ORD' 339456,
                                device '/dev/rD1F04V2ORD' 339456,
                                device '/dev/rD1F04V3ORD' 339456,
                                device '/dev/rD1F05V1ORD' 339456,
                                device '/dev/rD1F05V2ORD' 339456,
                                device '/dev/rD1F05V3ORD' 339456,
                                device '/dev/rD1F06V1ORD' 339456,
                                device '/dev/rD1F06V2ORD' 339456,
                                device '/dev/rD1F06V3ORD' 339456,
                                device '/dev/rD1F07V1ORD' 339456,
                                device '/dev/rD1F07V2ORD' 339456,
                                device '/dev/rD1F07V3ORD' 339456,
                                device '/dev/rD1F08V1ORD' 339456,
                                device '/dev/rD1F08V2ORD' 339456,
                                device '/dev/rD1F08V3ORD' 339456,
                                device '/dev/rD1F09V1ORD' 339456,
                                device '/dev/rD1F09V2ORD' 339456,
                                device '/dev/rD1F09V3ORD' 339456,
                                device '/dev/rD1F10V1ORD' 339456,
                                device '/dev/rD1F10V2ORD' 339456,
                                device '/dev/rD1F10V3ORD' 339456,
                                device '/dev/rD1F11V1ORD' 339456,
                                device '/dev/rD1F11V2ORD' 339456,
                                device '/dev/rD1F11V3ORD' 339456,
                                device '/dev/rD1F12V1ORD' 339456,
                                device '/dev/rD1F12V2ORD' 339456,
                                device '/dev/rD1F12V3ORD' 339456,
                                device '/dev/rD1F13V1ORD' 339456,
                                device '/dev/rD1F13V2ORD' 339456,
                                device '/dev/rD1F13V3ORD' 339456
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;


-- now creating TS for is_order_01 of D1
drop tablespace is_order_01;
create regular tablespace is_order_01 pagesize 8K
                managed by database
                using
               (
                                device '/dev/rD1F01V1ORDI' 292608,
                                device '/dev/rD1F01V2ORDI' 292608,
                                device '/dev/rD1F01V3ORDI' 292608,
                                device '/dev/rD1F02V1ORDI' 292608,
                                device '/dev/rD1F02V2ORDI' 292608,
                                device '/dev/rD1F02V3ORDI' 292608,
                                device '/dev/rD1F03V1ORDI' 292608,
                                device '/dev/rD1F03V2ORDI' 292608,
                                device '/dev/rD1F03V3ORDI' 292608,
                                device '/dev/rD1F04V1ORDI' 292608,
                                device '/dev/rD1F04V2ORDI' 292608,
                                device '/dev/rD1F04V3ORDI' 292608,
                                device '/dev/rD1F05V1ORDI' 292608,
                                device '/dev/rD1F05V2ORDI' 292608,
                                device '/dev/rD1F05V3ORDI' 292608,
                                device '/dev/rD1F06V1ORDI' 292608,
                                device '/dev/rD1F06V2ORDI' 292608,
                                device '/dev/rD1F06V3ORDI' 292608,
                                device '/dev/rD1F07V1ORDI' 292608,
                                device '/dev/rD1F07V2ORDI' 292608,
                                device '/dev/rD1F07V3ORDI' 292608,
                                device '/dev/rD1F08V1ORDI' 292608,
                                device '/dev/rD1F08V2ORDI' 292608,
                                device '/dev/rD1F08V3ORDI' 292608,
                                device '/dev/rD1F09V1ORDI' 292608,
                                device '/dev/rD1F09V2ORDI' 292608,
                                device '/dev/rD1F09V3ORDI' 292608,
                                device '/dev/rD1F10V1ORDI' 292608,
                                device '/dev/rD1F10V2ORDI' 292608,
                                device '/dev/rD1F10V3ORDI' 292608,
                                device '/dev/rD1F11V1ORDI' 292608,
                                device '/dev/rD1F11V2ORDI' 292608,
                                device '/dev/rD1F11V3ORDI' 292608,
                                device '/dev/rD1F12V1ORDI' 292608,
                                device '/dev/rD1F12V2ORDI' 292608,
                                device '/dev/rD1F12V3ORDI' 292608,
                                device '/dev/rD1F13V1ORDI' 292608,
                                device '/dev/rD1F13V2ORDI' 292608,
                                device '/dev/rD1F13V3ORDI' 292608
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_01 of D1

drop tablespace ts_orderline_01;
create regular tablespace ts_orderline_01 pagesize 8K
                managed by database
                using
               (
                                device '/dev/rD1F01V1ORL1' 9181440
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_02 of D1

drop tablespace ts_orderline_02;
create regular tablespace ts_orderline_02 pagesize 8K
                managed by database
                using
               (
                                device '/dev/rD1F01V2ORL2' 9181440
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_03 of D1

drop tablespace ts_orderline_03;
create regular tablespace ts_orderline_03 pagesize 8K
                managed by database
                using
               (
                                device '/dev/rD1F01V3ORL3' 9181440
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_04 of D1

drop tablespace ts_orderline_04;
create regular tablespace ts_orderline_04 pagesize 8K
                managed by database
                using
               (
                                device '/dev/rD1F02V1ORL4' 9181440
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_05 of D1

drop tablespace ts_orderline_05;
create regular tablespace ts_orderline_05 pagesize 8K
                managed by database
                using
               (
                                device '/dev/rD1F02V2ORL5' 9181440
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_06 of D1

drop tablespace ts_orderline_06;
create regular tablespace ts_orderline_06 pagesize 8K
                managed by database
                using
               (
                                device '/dev/rD1F02V3ORL6' 9181440
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_07 of D1

drop tablespace ts_orderline_07;
create regular tablespace ts_orderline_07 pagesize 8K
                managed by database
                using
               (
                                device '/dev/rD1F03V1ORL7' 9181440
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;
```

```
-- now creating TS for ts_orderline_08 of D1

drop tablespace ts_orderline_08;
create regular tablespace ts_orderline_08 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F03V2ORL8' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_09 of D1

drop tablespace ts_orderline_09;
create regular tablespace ts_orderline_09 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F03V3ORL9' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_10 of D1

drop tablespace ts_orderline_10;
create regular tablespace ts_orderline_10 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F04V1ORL10' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_11 of D1

drop tablespace ts_orderline_11;
create regular tablespace ts_orderline_11 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F04V2ORL11' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_12 of D1

drop tablespace ts_orderline_12;
create regular tablespace ts_orderline_12 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F04V3ORL12' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_13 of D1

drop tablespace ts_orderline_13;
create regular tablespace ts_orderline_13 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F05V1ORL13' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_14 of D1

drop tablespace ts_orderline_14;
create regular tablespace ts_orderline_14 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F05V2ORL14' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_15 of D1

drop tablespace ts_orderline_15;
create regular tablespace ts_orderline_15 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F05V3ORL15' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_16 of D1

drop tablespace ts_orderline_16;
create regular tablespace ts_orderline_16 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F06V1ORL16' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_17 of D1

drop tablespace ts_orderline_17;
create regular tablespace ts_orderline_17 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F06V2ORL17' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_18 of D1

drop tablespace ts_orderline_18;
create regular tablespace ts_orderline_18 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F06V3ORL18' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_19 of D1

drop tablespace ts_orderline_19;
create regular tablespace ts_orderline_19 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F07V1ORL19' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_20 of D1

drop tablespace ts_orderline_20;
create regular tablespace ts_orderline_20 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F07V2ORL20' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_21 of D1

drop tablespace ts_orderline_21;
create regular tablespace ts_orderline_21 pagesize 8K
                managed by database
                using
               (
                              device '/dev/rD1F07V3ORL21' 9181440
                             )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_22 of D1
```

```
drop tablespace ts_orderline_22;
create regular tablespace ts_orderline_22 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F08V1ORL22' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_23 of D1

drop tablespace ts_orderline_23;
create regular tablespace ts_orderline_23 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F08V2ORL23' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_24 of D1

drop tablespace ts_orderline_24;
create regular tablespace ts_orderline_24 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F08V3ORL24' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_25 of D1

drop tablespace ts_orderline_25;
create regular tablespace ts_orderline_25 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F09V1ORL25' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_26 of D1

drop tablespace ts_orderline_26;
create regular tablespace ts_orderline_26 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F09V2ORL26' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_27 of D1

drop tablespace ts_orderline_27;
create regular tablespace ts_orderline_27 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F09V3ORL27' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_28 of D1

drop tablespace ts_orderline_28;
create regular tablespace ts_orderline_28 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F10V1ORL28' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_29 of D1

drop tablespace ts_orderline_29;
create regular tablespace ts_orderline_29 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F10V2ORL29' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_30 of D1

drop tablespace ts_orderline_30;
create regular tablespace ts_orderline_30 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F10V3ORL30' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_31 of D1

drop tablespace ts_orderline_31;
create regular tablespace ts_orderline_31 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F11V1ORL31' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_32 of D1

drop tablespace ts_orderline_32;
create regular tablespace ts_orderline_32 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F11V2ORL32' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_33 of D1

drop tablespace ts_orderline_33;
create regular tablespace ts_orderline_33 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F11V3ORL33' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_34 of D1

drop tablespace ts_orderline_34;
create regular tablespace ts_orderline_34 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F12V1ORL34' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_35 of D1

drop tablespace ts_orderline_35;
create regular tablespace ts_orderline_35 pagesize 8K
                managed by database
                using
                (
                                device '/dev/rD1F12V2ORL35' 9181440
                                )
                extentsize 256
                prefetchsize 512
                bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_36 of D1

drop tablespace ts_orderline_36;
```

```
create regular tablespace ts_orderline_36 pagesize 8K
                managed by database
                using
               (
                                device '/dev/rD1F12V3ORL36' 9181440
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_37 of D1

drop tablespace ts_orderline_37;
create regular tablespace ts_orderline_37 pagesize 8K
                managed by database
                using
               (
                                device '/dev/rD1F13V1ORL37' 9181440
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_38 of D1

drop tablespace ts_orderline_38;
create regular tablespace ts_orderline_38 pagesize 8K
                managed by database
                using
               (
                                device '/dev/rD1F13V2ORL38' 9181440
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;

-- now creating TS for ts_orderline_39 of D1

drop tablespace ts_orderline_39;
create regular tablespace ts_orderline_39 pagesize 8K
                managed by database
                using
               (
                                device '/dev/rD1F13V3ORL39' 9181440
                               )
               extentsize 256
               prefetchsize 512
               bufferpool ibmdefaultbp8K;
commit;


-- now creating TS for ts_neword_01 of D1
drop tablespace ts_neword_01;
create regular tablespace ts_neword_01 pagesize 4K
                managed by database
                using
               (
                                device '/dev/rD1F01V1NOR' 181760,
                                device '/dev/rD1F01V2NOR' 181760,
                                device '/dev/rD1F01V3NOR' 181760,
                                device '/dev/rD1F02V1NOR' 181760,
                                device '/dev/rD1F02V2NOR' 181760,
                                device '/dev/rD1F02V3NOR' 181760,
                                device '/dev/rD1F03V1NOR' 181760,
                                device '/dev/rD1F03V2NOR' 181760,
                                device '/dev/rD1F03V3NOR' 181760,
                                device '/dev/rD1F04V1NOR' 181760,
                                device '/dev/rD1F04V2NOR' 181760,
                                device '/dev/rD1F04V3NOR' 181760,
                                device '/dev/rD1F05V1NOR' 181760,
                                device '/dev/rD1F05V2NOR' 181760,
                                device '/dev/rD1F05V3NOR' 181760,
                                device '/dev/rD1F06V1NOR' 181760,
                                device '/dev/rD1F06V2NOR' 181760,
                                device '/dev/rD1F06V3NOR' 181760,
                                device '/dev/rD1F07V1NOR' 181760,
                                device '/dev/rD1F07V2NOR' 181760,
                                device '/dev/rD1F07V3NOR' 181760,
                                device '/dev/rD1F08V1NOR' 181760,
                                device '/dev/rD1F08V2NOR' 181760,
                                device '/dev/rD1F08V3NOR' 181760,
                                device '/dev/rD1F09V1NOR' 181760,
                                device '/dev/rD1F09V2NOR' 181760,
                                device '/dev/rD1F09V3NOR' 181760,
                                device '/dev/rD1F10V1NOR' 181760,
                                device '/dev/rD1F10V2NOR' 181760,
                                device '/dev/rD1F10V3NOR' 181760,
                                device '/dev/rD1F11V1NOR' 181760,
                                device '/dev/rD1F11V2NOR' 181760,
                                device '/dev/rD1F11V3NOR' 181760,
                                device '/dev/rD1F12V1NOR' 181760,
                                device '/dev/rD1F12V2NOR' 181760,
                                device '/dev/rD1F12V3NOR' 181760,
                                device '/dev/rD1F13V1NOR' 181760,
                                device '/dev/rD1F13V2NOR' 181760,
                                device '/dev/rD1F13V3NOR' 181760
                               )
               extentsize 256
               prefetchsize 512;
commit;


-- now creating TS for is_neword_01 of D1
drop tablespace is_neword_01;
create regular tablespace is_neword_01 pagesize 4K
                managed by database
                using
               (
                                device '/dev/rD1F01V1NORI' 181760,
                                device '/dev/rD1F01V2NORI' 181760,
                                device '/dev/rD1F01V3NORI' 181760,
                                device '/dev/rD1F02V1NORI' 181760,
                                device '/dev/rD1F02V2NORI' 181760,
                                device '/dev/rD1F02V3NORI' 181760,
                                device '/dev/rD1F03V1NORI' 181760,
                                device '/dev/rD1F03V2NORI' 181760,
                                device '/dev/rD1F03V3NORI' 181760,
                                device '/dev/rD1F04V1NORI' 181760,
                                device '/dev/rD1F04V2NORI' 181760,
                                device '/dev/rD1F04V3NORI' 181760,
                                device '/dev/rD1F05V1NORI' 181760,
                                device '/dev/rD1F05V2NORI' 181760,
                                device '/dev/rD1F05V3NORI' 181760,
                                device '/dev/rD1F06V1NORI' 181760,
                                device '/dev/rD1F06V2NORI' 181760,
                                device '/dev/rD1F06V3NORI' 181760,
                                device '/dev/rD1F07V1NORI' 181760,
                                device '/dev/rD1F07V2NORI' 181760,
                                device '/dev/rD1F07V3NORI' 181760,
                                device '/dev/rD1F08V1NORI' 181760,
                                device '/dev/rD1F08V2NORI' 181760,
                                device '/dev/rD1F08V3NORI' 181760,
                                device '/dev/rD1F09V1NORI' 181760,
                                device '/dev/rD1F09V2NORI' 181760,
                                device '/dev/rD1F09V3NORI' 181760,
                                device '/dev/rD1F10V1NORI' 181760,
                                device '/dev/rD1F10V2NORI' 181760,
                                device '/dev/rD1F10V3NORI' 181760,
                                device '/dev/rD1F11V1NORI' 181760,
                                device '/dev/rD1F11V2NORI' 181760,
                                device '/dev/rD1F11V3NORI' 181760,
                                device '/dev/rD1F12V1NORI' 181760,
                                device '/dev/rD1F12V2NORI' 181760,
                                device '/dev/rD1F12V3NORI' 181760,
                                device '/dev/rD1F13V1NORI' 181760,
                                device '/dev/rD1F13V2NORI' 181760,
                                device '/dev/rD1F13V3NORI' 181760
                               )
               extentsize 256
               prefetchsize 512;
commit;

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           319
#define C_OL_I_ID        3849
#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|%d|%s|%s|%s|%s|%s|%s|%s|%s|%s|%s|%d|%ld\n";
char fmtCust[]    = "%d|%s|%s|%s|%s|%lld|%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|%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

```
#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 60900
#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
#define MASTER_NUM10 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"
#elif MASTER_NUM10
MASTER "master10"
#endif
/*---- SUT ----------------------------------------------*/
SUT="oak"
/*-------------------------------------------------------*/
LASTC=88
MEASUREMENT="1"
WAREHOUSES=60900
/*----- 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
#elif MASTER_NUM10
SLAVES driver55a, driver55b, driver55c, driver55d, driver55e, driver55f, driver56a, driver56b, driver56c,
driver56d, driver56e, driver56f, driver57a, driver57b, driver57c, driver57d, driver57e, driver57f, driver58a,
driver58b, driver58c, driver58d, driver58e, driver58f, driver59a, driver59b, driver59c, driver59d, driver59e,
driver59f, driver60a, driver60b, driver60c, driver60d, driver60e, driver60f
#endif

/* ---- CLIENTS ----------------------------------------------*/
#if MASTER_NUM1
MAIN_CLIENT = client1
CLIENT_REAL = "client1 client2 client3"
#elif MASTER_NUM2
MAIN_CLIENT = client4
CLIENT_REAL = "client4 client5 client6"
#elif MASTER_NUM3
MAIN_CLIENT = client7
CLIENT_REAL = "client7 client8 client9"
#elif MASTER_NUM4
MAIN_CLIENT = client10
CLIENT_REAL = "client10 client11 client12"
#elif MASTER_NUM5
MAIN_CLIENT = client13
CLIENT_REAL = "client13 client14 client15"
#elif MASTER_NUM6
MAIN_CLIENT = client16
CLIENT_REAL = "client16 client17 client18"
#elif MASTER_NUM7
MAIN_CLIENT = client19
CLIENT_REAL = "client19 client20 client21"
#elif MASTER_NUM8
MAIN_CLIENT = client22
CLIENT_REAL = "client22 client23 client24"
#elif MASTER_NUM9
MAIN_CLIENT = client25
CLIENT_REAL = "client25 client26 client27"
#elif MASTER_NUM10
MAIN_CLIENT = client28
CLIENT_REAL = "client28 client29 client30"
#endif

/*---- more client ctuff ------------------------------------*/
#if MASTER_NUM1
CLIENT 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

#elif MASTER_NUM10
CLIENT client28a tpcc tpcc
CLIENT client28b tpcc tpcc
CLIENT client28c tpcc tpcc
CLIENT client28d tpcc tpcc
CLIENT client28e tpcc tpcc
CLIENT client28f tpcc tpcc
CLIENT client28g tpcc tpcc
CLIENT client28h tpcc tpcc
CLIENT client28i tpcc tpcc
CLIENT client28j tpcc tpcc
CLIENT client28k tpcc tpcc
CLIENT client28l tpcc tpcc
CLIENT client28m tpcc tpcc
CLIENT client28n tpcc tpcc
CLIENT client28o tpcc tpcc
CLIENT client28p tpcc tpcc
CLIENT client28q tpcc tpcc
CLIENT client28r tpcc tpcc
CLIENT client28s tpcc tpcc
CLIENT client28t tpcc tpcc
CLIENT client28u tpcc tpcc
CLIENT client28v tpcc tpcc
CLIENT client28w tpcc tpcc
CLIENT client28x tpcc tpcc
CLIENT client28y tpcc tpcc
CLIENT client28z tpcc tpcc
CLIENT client28aa tpcc tpcc
CLIENT client28ab tpcc tpcc

CLIENT client29a tpcc tpcc
CLIENT client29b tpcc tpcc
CLIENT client29c tpcc tpcc
CLIENT client29d tpcc tpcc
CLIENT client29e tpcc tpcc
CLIENT client29f tpcc tpcc
CLIENT client29g tpcc tpcc
CLIENT client29h tpcc tpcc
CLIENT client29i tpcc tpcc
CLIENT client29j tpcc tpcc
CLIENT client29k tpcc tpcc
CLIENT client29l tpcc tpcc
CLIENT client29m tpcc tpcc
CLIENT client29n tpcc tpcc
CLIENT client29o tpcc tpcc
CLIENT client29p tpcc tpcc
CLIENT client29q tpcc tpcc
CLIENT client29r tpcc tpcc
CLIENT client29s tpcc tpcc
CLIENT client29t tpcc tpcc
CLIENT client29u tpcc tpcc
CLIENT client29v tpcc tpcc
CLIENT client29w tpcc tpcc
CLIENT client29x tpcc tpcc
CLIENT client29y tpcc tpcc
CLIENT client29z tpcc tpcc
CLIENT client29aa tpcc tpcc
CLIENT client29ab tpcc tpcc
```

```
CLIENT client30a tpcc tpcc
CLIENT client30b tpcc tpcc
CLIENT client30c tpcc tpcc
CLIENT client30d tpcc tpcc
CLIENT client30e tpcc tpcc
CLIENT client30f tpcc tpcc
CLIENT client30g tpcc tpcc
CLIENT client30h tpcc tpcc
CLIENT client30i tpcc tpcc
CLIENT client30j tpcc tpcc
CLIENT client30k tpcc tpcc
CLIENT client30l tpcc tpcc
CLIENT client30m tpcc tpcc
CLIENT client30n tpcc tpcc
CLIENT client30o tpcc tpcc
CLIENT client30p tpcc tpcc
CLIENT client30q tpcc tpcc
CLIENT client30r tpcc tpcc
CLIENT client30s tpcc tpcc
CLIENT client30t tpcc tpcc
CLIENT client30u tpcc tpcc
CLIENT client30v tpcc tpcc
CLIENT client30w tpcc tpcc
CLIENT client30x tpcc tpcc
CLIENT client30y tpcc tpcc
CLIENT client30z tpcc tpcc
CLIENT client30aa tpcc tpcc
CLIENT client30ab 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 driver1c
SOCKET_NETWORK socket58   6557 driver1d
SOCKET_NETWORK socket59   6558 driver1e
SOCKET_NETWORK socket60   6559 driver1f
SOCKET_NETWORK socket61   6560 driver1a
SOCKET_NETWORK socket62   6561 driver1b
SOCKET_NETWORK socket63   6562 driver1c
SOCKET_NETWORK socket64   6563 driver1d
SOCKET_NETWORK socket65   6564 driver1e
SOCKET_NETWORK socket66   6565 driver1f
SOCKET_NETWORK socket67   6566 driver1a
SOCKET_NETWORK socket68   6567 driver1b
SOCKET_NETWORK socket69   6568 driver1c
SOCKET_NETWORK socket70   6569 driver1d
SOCKET_NETWORK socket71   6570 driver2a
SOCKET_NETWORK socket72   6571 driver2b
SOCKET_NETWORK socket73   6572 driver2c
SOCKET_NETWORK socket74   6573 driver2d
SOCKET_NETWORK socket75   6574 driver2e
SOCKET_NETWORK socket76   6575 driver2f
SOCKET_NETWORK socket77   6576 driver2a
SOCKET_NETWORK socket78   6577 driver2b
SOCKET_NETWORK socket79   6578 driver2c
SOCKET_NETWORK socket80   6579 driver2d
SOCKET_NETWORK socket81   6580 driver2e
SOCKET_NETWORK socket82   6581 driver2f
SOCKET_NETWORK socket83   6582 driver2a
SOCKET_NETWORK socket84   6583 driver2b
SOCKET_NETWORK socket85   6584 driver2c
SOCKET_NETWORK socket86   6585 driver2d
SOCKET_NETWORK socket87   6586 driver2e
SOCKET_NETWORK socket88   6587 driver2f
SOCKET_NETWORK socket89   6588 driver2a
SOCKET_NETWORK socket90   6589 driver2b
SOCKET_NETWORK socket91   6590 driver2c
SOCKET_NETWORK socket92   6591 driver2d
SOCKET_NETWORK socket93   6592 driver2e
SOCKET_NETWORK socket94   6593 driver2f
SOCKET_NETWORK socket95   6594 driver2a
SOCKET_NETWORK socket96   6595 driver2b
SOCKET_NETWORK socket97   6596 driver2c
SOCKET_NETWORK socket98   6597 driver2d
SOCKET_NETWORK socket99   6598 driver2e
SOCKET_NETWORK socket100   6599 driver2f
SOCKET_NETWORK socket101   6600 driver2a
SOCKET_NETWORK socket102   6601 driver2b
SOCKET_NETWORK socket103   6602 driver2c
SOCKET_NETWORK socket104   6603 driver2d
SOCKET_NETWORK socket105   6604 driver2e
SOCKET_NETWORK socket106   6605 driver2f
SOCKET_NETWORK socket107   6606 driver2a
SOCKET_NETWORK socket108   6607 driver2b
SOCKET_NETWORK socket109   6608 driver2c
SOCKET_NETWORK socket110   6609 driver2d
SOCKET_NETWORK socket111   6610 driver2e
SOCKET_NETWORK socket112   6611 driver2f
SOCKET_NETWORK socket113   6612 driver2a
SOCKET_NETWORK socket114   6613 driver2b
SOCKET_NETWORK socket115   6614 driver2c
SOCKET_NETWORK socket116   6615 driver2d
SOCKET_NETWORK socket117   6616 driver2e
SOCKET_NETWORK socket118   6617 driver2f
SOCKET_NETWORK socket119   6618 driver2a
SOCKET_NETWORK socket120   6619 driver2b
SOCKET_NETWORK socket121   6620 driver2c
SOCKET_NETWORK socket122   6621 driver2d
SOCKET_NETWORK socket123   6622 driver2e
SOCKET_NETWORK socket124   6623 driver2f
SOCKET_NETWORK socket125   6624 driver2a
SOCKET_NETWORK socket126   6625 driver2b
SOCKET_NETWORK socket127   6626 driver2c
SOCKET_NETWORK socket128   6627 driver2d
SOCKET_NETWORK socket129   6628 driver2e
SOCKET_NETWORK socket130   6629 driver2f
SOCKET_NETWORK socket131   6630 driver2a
SOCKET_NETWORK socket132   6631 driver2b
SOCKET_NETWORK socket133   6632 driver2c
SOCKET_NETWORK socket134   6633 driver2d
SOCKET_NETWORK socket135   6634 driver2e
SOCKET_NETWORK socket136   6635 driver2f
SOCKET_NETWORK socket137   6636 driver2a
SOCKET_NETWORK socket138   6637 driver2b
SOCKET_NETWORK socket139   6638 driver2c
SOCKET_NETWORK socket140   6639 driver2d
SOCKET_NETWORK socket141   6500 driver3a
SOCKET_NETWORK socket142   6501 driver3b
SOCKET_NETWORK socket143   6502 driver3c
SOCKET_NETWORK socket144   6503 driver3d
SOCKET_NETWORK socket145   6504 driver3e
SOCKET_NETWORK socket146   6505 driver3f
SOCKET_NETWORK socket147   6506 driver3a
SOCKET_NETWORK socket148   6507 driver3b
SOCKET_NETWORK socket149   6508 driver3c
SOCKET_NETWORK socket150   6509 driver3d
SOCKET_NETWORK socket151   6510 driver3e
SOCKET_NETWORK socket152   6511 driver3f
SOCKET_NETWORK socket153   6512 driver3a
SOCKET_NETWORK socket154   6513 driver3b
SOCKET_NETWORK socket155   6514 driver3c
SOCKET_NETWORK socket156   6515 driver3d
SOCKET_NETWORK socket157   6516 driver3e
SOCKET_NETWORK socket158   6517 driver3f
SOCKET_NETWORK socket159   6518 driver3a
SOCKET_NETWORK socket160   6519 driver3b
SOCKET_NETWORK socket161   6520 driver3c
SOCKET_NETWORK socket162   6521 driver3d
SOCKET_NETWORK socket163   6522 driver3e
```

SOCKET_NETWORK socket164 6523 driver3f
SOCKET_NETWORK socket165 6524 driver3a
SOCKET_NETWORK socket166 6525 driver3b
SOCKET_NETWORK socket167 6526 driver3c
SOCKET_NETWORK socket168 6527 driver3d
SOCKET_NETWORK socket169 6528 driver3e
SOCKET_NETWORK socket170 6529 driver3f
SOCKET_NETWORK socket171 6530 driver3a
SOCKET_NETWORK socket172 6531 driver3b
SOCKET_NETWORK socket173 6532 driver3c
SOCKET_NETWORK socket174 6533 driver3d
SOCKET_NETWORK socket175 6534 driver3e
SOCKET_NETWORK socket176 6535 driver3f
SOCKET_NETWORK socket177 6536 driver3a
SOCKET_NETWORK socket178 6537 driver3b
SOCKET_NETWORK socket179 6538 driver3c
SOCKET_NETWORK socket180 6539 driver3d
SOCKET_NETWORK socket181 6540 driver3e
SOCKET_NETWORK socket182 6541 driver3f
SOCKET_NETWORK socket183 6542 driver3a
SOCKET_NETWORK socket184 6543 driver3b
SOCKET_NETWORK socket185 6544 driver3c
SOCKET_NETWORK socket186 6545 driver3d
SOCKET_NETWORK socket187 6546 driver3e
SOCKET_NETWORK socket188 6547 driver3f
SOCKET_NETWORK socket189 6548 driver3a
SOCKET_NETWORK socket190 6549 driver3b
SOCKET_NETWORK socket191 6550 driver3c
SOCKET_NETWORK socket192 6551 driver3d
SOCKET_NETWORK socket193 6552 driver3e
SOCKET_NETWORK socket194 6553 driver3f
SOCKET_NETWORK socket195 6554 driver3a
SOCKET_NETWORK socket196 6555 driver3b
SOCKET_NETWORK socket197 6556 driver3c
SOCKET_NETWORK socket198 6557 driver3d
SOCKET_NETWORK socket199 6558 driver3e
SOCKET_NETWORK socket200 6559 driver3f
SOCKET_NETWORK socket201 6560 driver3a
SOCKET_NETWORK socket202 6561 driver3b
SOCKET_NETWORK socket203 6562 driver3c
SOCKET_NETWORK socket204 6563 driver3d
SOCKET_NETWORK socket205 6564 driver3e
SOCKET_NETWORK socket206 6565 driver3f
SOCKET_NETWORK socket207 6566 driver3a
SOCKET_NETWORK socket208 6567 driver3b
SOCKET_NETWORK socket209 6568 driver3c
SOCKET_NETWORK socket210 6569 driver3d
SOCKET_NETWORK socket211 6570 driver4a
SOCKET_NETWORK socket212 6571 driver4b
SOCKET_NETWORK socket213 6572 driver4c
SOCKET_NETWORK socket214 6573 driver4d
SOCKET_NETWORK socket215 6574 driver4e
SOCKET_NETWORK socket216 6575 driver4f
SOCKET_NETWORK socket217 6576 driver4a
SOCKET_NETWORK socket218 6577 driver4b
SOCKET_NETWORK socket219 6578 driver4c
SOCKET_NETWORK socket220 6579 driver4d
SOCKET_NETWORK socket221 6580 driver4e
SOCKET_NETWORK socket222 6581 driver4f
SOCKET_NETWORK socket223 6582 driver4a
SOCKET_NETWORK socket224 6583 driver4b
SOCKET_NETWORK socket225 6584 driver4c
SOCKET_NETWORK socket226 6585 driver4d
SOCKET_NETWORK socket227 6586 driver4e
SOCKET_NETWORK socket228 6587 driver4f
SOCKET_NETWORK socket229 6588 driver4a
SOCKET_NETWORK socket230 6589 driver4b
SOCKET_NETWORK socket231 6590 driver4c
SOCKET_NETWORK socket232 6591 driver4d
SOCKET_NETWORK socket233 6592 driver4e
SOCKET_NETWORK socket234 6593 driver4f
SOCKET_NETWORK socket235 6594 driver4a
SOCKET_NETWORK socket236 6595 driver4b
SOCKET_NETWORK socket237 6596 driver4c
SOCKET_NETWORK socket238 6597 driver4d
SOCKET_NETWORK socket239 6598 driver4e
SOCKET_NETWORK socket240 6599 driver4f
SOCKET_NETWORK socket241 6600 driver4a
SOCKET_NETWORK socket242 6601 driver4b
SOCKET_NETWORK socket243 6602 driver4c
SOCKET_NETWORK socket244 6603 driver4d
SOCKET_NETWORK socket245 6604 driver4e
SOCKET_NETWORK socket246 6605 driver4f
SOCKET_NETWORK socket247 6606 driver4a
SOCKET_NETWORK socket248 6607 driver4b
SOCKET_NETWORK socket249 6608 driver4c
SOCKET_NETWORK socket250 6609 driver4d
SOCKET_NETWORK socket251 6610 driver4e
SOCKET_NETWORK socket252 6611 driver4f
SOCKET_NETWORK socket253 6612 driver4a
SOCKET_NETWORK socket254 6613 driver4b
SOCKET_NETWORK socket255 6614 driver4c
SOCKET_NETWORK socket256 6615 driver4d
SOCKET_NETWORK socket257 6616 driver4e
SOCKET_NETWORK socket258 6617 driver4f
SOCKET_NETWORK socket259 6618 driver4a
SOCKET_NETWORK socket260 6619 driver4b
SOCKET_NETWORK socket261 6620 driver4c
SOCKET_NETWORK socket262 6621 driver4d
SOCKET_NETWORK socket263 6622 driver4e
SOCKET_NETWORK socket264 6623 driver4f
SOCKET_NETWORK socket265 6624 driver4a
SOCKET_NETWORK socket266 6625 driver4b
SOCKET_NETWORK socket267 6626 driver4c
SOCKET_NETWORK socket268 6627 driver4d
SOCKET_NETWORK socket269 6628 driver4e
SOCKET_NETWORK socket270 6629 driver4f
SOCKET_NETWORK socket271 6630 driver4a
SOCKET_NETWORK socket272 6631 driver4b
SOCKET_NETWORK socket273 6632 driver4c
SOCKET_NETWORK socket274 6633 driver4d
SOCKET_NETWORK socket275 6634 driver4e
SOCKET_NETWORK socket276 6635 driver4f
SOCKET_NETWORK socket277 6636 driver4a
SOCKET_NETWORK socket278 6637 driver4b
SOCKET_NETWORK socket279 6638 driver4c
SOCKET_NETWORK socket280 6639 driver4d
SOCKET_NETWORK socket281 6500 driver5a
SOCKET_NETWORK socket282 6501 driver5b
SOCKET_NETWORK socket283 6502 driver5c
SOCKET_NETWORK socket284 6503 driver5d
SOCKET_NETWORK socket285 6504 driver5e
SOCKET_NETWORK socket286 6505 driver5f
SOCKET_NETWORK socket287 6506 driver5a
SOCKET_NETWORK socket288 6507 driver5b
SOCKET_NETWORK socket289 6508 driver5c
SOCKET_NETWORK socket290 6509 driver5d
SOCKET_NETWORK socket291 6510 driver5e
SOCKET_NETWORK socket292 6511 driver5f
SOCKET_NETWORK socket293 6512 driver5a
SOCKET_NETWORK socket294 6513 driver5b
SOCKET_NETWORK socket295 6514 driver5c
SOCKET_NETWORK socket296 6515 driver5d
SOCKET_NETWORK socket297 6516 driver5e
SOCKET_NETWORK socket298 6517 driver5f
SOCKET_NETWORK socket299 6518 driver5a
SOCKET_NETWORK socket300 6519 driver5b
SOCKET_NETWORK socket301 6520 driver5c
SOCKET_NETWORK socket302 6521 driver5d
SOCKET_NETWORK socket303 6522 driver5e
SOCKET_NETWORK socket304 6523 driver5f
SOCKET_NETWORK socket305 6524 driver5a
SOCKET_NETWORK socket306 6525 driver5b
SOCKET_NETWORK socket307 6526 driver5c
SOCKET_NETWORK socket308 6527 driver5d
SOCKET_NETWORK socket309 6528 driver5e
SOCKET_NETWORK socket310 6529 driver5f
SOCKET_NETWORK socket311 6530 driver5a
SOCKET_NETWORK socket312 6531 driver5b
SOCKET_NETWORK socket313 6532 driver5c
SOCKET_NETWORK socket314 6533 driver5d
SOCKET_NETWORK socket315 6534 driver5e
SOCKET_NETWORK socket316 6535 driver5f
SOCKET_NETWORK socket317 6536 driver5a
SOCKET_NETWORK socket318 6537 driver5b
SOCKET_NETWORK socket319 6538 driver5c
SOCKET_NETWORK socket320 6539 driver5d
SOCKET_NETWORK socket321 6540 driver5e
SOCKET_NETWORK socket322 6541 driver5f
SOCKET_NETWORK socket323 6542 driver5a
SOCKET_NETWORK socket324 6543 driver5b
SOCKET_NETWORK socket325 6544 driver5c
SOCKET_NETWORK socket326 6545 driver5d
SOCKET_NETWORK socket327 6546 driver5e
SOCKET_NETWORK socket328 6547 driver5f
SOCKET_NETWORK socket329 6548 driver5a
SOCKET_NETWORK socket330 6549 driver5b
SOCKET_NETWORK socket331 6550 driver5c
SOCKET_NETWORK socket332 6551 driver5d
SOCKET_NETWORK socket333 6552 driver5e
SOCKET_NETWORK socket334 6553 driver5f
SOCKET_NETWORK socket335 6554 driver5a
SOCKET_NETWORK socket336 6555 driver5b
SOCKET_NETWORK socket337 6556 driver5c
SOCKET_NETWORK socket338 6557 driver5d
SOCKET_NETWORK socket339 6558 driver5e
SOCKET_NETWORK socket340 6559 driver5f
SOCKET_NETWORK socket341 6560 driver5a
SOCKET_NETWORK socket342 6561 driver5b
SOCKET_NETWORK socket343 6562 driver5c
SOCKET_NETWORK socket344 6563 driver5d
SOCKET_NETWORK socket345 6564 driver5e
SOCKET_NETWORK socket346 6565 driver5f
SOCKET_NETWORK socket347 6566 driver5a
SOCKET_NETWORK socket348 6567 driver5b
SOCKET_NETWORK socket349 6568 driver5c
SOCKET_NETWORK socket350 6569 driver5d
SOCKET_NETWORK socket351 6570 driver6a
SOCKET_NETWORK socket352 6571 driver6b
SOCKET_NETWORK socket353 6572 driver6c
SOCKET_NETWORK socket354 6573 driver6d
SOCKET_NETWORK socket355 6574 driver6e
SOCKET_NETWORK socket356 6575 driver6f
SOCKET_NETWORK socket357 6576 driver6a
SOCKET_NETWORK socket358 6577 driver6b
SOCKET_NETWORK socket359 6578 driver6c
SOCKET_NETWORK socket360 6579 driver6d
SOCKET_NETWORK socket361 6580 driver6e

```
SOCKET_NETWORK socket362  6581 driver6f
SOCKET_NETWORK socket363  6582 driver6a
SOCKET_NETWORK socket364  6583 driver6b
SOCKET_NETWORK socket365  6584 driver6c
SOCKET_NETWORK socket366  6585 driver6d
SOCKET_NETWORK socket367  6586 driver6e
SOCKET_NETWORK socket368  6587 driver6f
SOCKET_NETWORK socket369  6588 driver6a
SOCKET_NETWORK socket370  6589 driver6b
SOCKET_NETWORK socket371  6590 driver6c
SOCKET_NETWORK socket372  6591 driver6d
SOCKET_NETWORK socket373  6592 driver6e
SOCKET_NETWORK socket374  6593 driver6f
SOCKET_NETWORK socket375  6594 driver6a
SOCKET_NETWORK socket376  6595 driver6b
SOCKET_NETWORK socket377  6596 driver6c
SOCKET_NETWORK socket378  6597 driver6d
SOCKET_NETWORK socket379  6598 driver6e
SOCKET_NETWORK socket380  6599 driver6f
SOCKET_NETWORK socket381  6600 driver6a
SOCKET_NETWORK socket382  6601 driver6b
SOCKET_NETWORK socket383  6602 driver6c
SOCKET_NETWORK socket384  6603 driver6d
SOCKET_NETWORK socket385  6604 driver6e
SOCKET_NETWORK socket386  6605 driver6f
SOCKET_NETWORK socket387  6606 driver6a
SOCKET_NETWORK socket388  6607 driver6b
SOCKET_NETWORK socket389  6608 driver6c
SOCKET_NETWORK socket390  6609 driver6d
SOCKET_NETWORK socket391  6610 driver6e
SOCKET_NETWORK socket392  6611 driver6f
SOCKET_NETWORK socket393  6612 driver6a
SOCKET_NETWORK socket394  6613 driver6b
SOCKET_NETWORK socket395  6614 driver6c
SOCKET_NETWORK socket396  6615 driver6d
SOCKET_NETWORK socket397  6616 driver6e
SOCKET_NETWORK socket398  6617 driver6f
SOCKET_NETWORK socket399  6618 driver6a
SOCKET_NETWORK socket400  6619 driver6b
SOCKET_NETWORK socket401  6620 driver6c
SOCKET_NETWORK socket402  6621 driver6d
SOCKET_NETWORK socket403  6622 driver6e
SOCKET_NETWORK socket404  6623 driver6f
SOCKET_NETWORK socket405  6624 driver6a
SOCKET_NETWORK socket406  6625 driver6b
SOCKET_NETWORK socket407  6626 driver6c
SOCKET_NETWORK socket408  6627 driver6d
SOCKET_NETWORK socket409  6628 driver6e
SOCKET_NETWORK socket410  6629 driver6f
SOCKET_NETWORK socket411  6630 driver6a
SOCKET_NETWORK socket412  6631 driver6b
SOCKET_NETWORK socket413  6632 driver6c
SOCKET_NETWORK socket414  6633 driver6d
SOCKET_NETWORK socket415  6634 driver6e
SOCKET_NETWORK socket416  6635 driver6f
SOCKET_NETWORK socket417  6636 driver6a
SOCKET_NETWORK socket418  6637 driver6b
SOCKET_NETWORK socket419  6638 driver6c
SOCKET_NETWORK socket420  6639 driver6d
#elif MASTER_NUM2
SOCKET_NETWORK socket421  6500 driver7a
SOCKET_NETWORK socket422  6501 driver7b
SOCKET_NETWORK socket423  6502 driver7c
SOCKET_NETWORK socket424  6503 driver7d
SOCKET_NETWORK socket425  6504 driver7e
SOCKET_NETWORK socket426  6505 driver7f
SOCKET_NETWORK socket427  6506 driver7a
SOCKET_NETWORK socket428  6507 driver7b
SOCKET_NETWORK socket429  6508 driver7c
SOCKET_NETWORK socket430  6509 driver7d
SOCKET_NETWORK socket431  6510 driver7e
SOCKET_NETWORK socket432  6511 driver7f
SOCKET_NETWORK socket433  6512 driver7a
SOCKET_NETWORK socket434  6513 driver7b
SOCKET_NETWORK socket435  6514 driver7c
SOCKET_NETWORK socket436  6515 driver7d
SOCKET_NETWORK socket437  6516 driver7e
SOCKET_NETWORK socket438  6517 driver7f
SOCKET_NETWORK socket439  6518 driver7a
SOCKET_NETWORK socket440  6519 driver7b
SOCKET_NETWORK socket441  6520 driver7c
SOCKET_NETWORK socket442  6521 driver7d
SOCKET_NETWORK socket443  6522 driver7e
SOCKET_NETWORK socket444  6523 driver7f
SOCKET_NETWORK socket445  6524 driver7a
SOCKET_NETWORK socket446  6525 driver7b
SOCKET_NETWORK socket447  6526 driver7c
SOCKET_NETWORK socket448  6527 driver7d
SOCKET_NETWORK socket449  6528 driver7e
SOCKET_NETWORK socket450  6529 driver7f
SOCKET_NETWORK socket451  6530 driver7a
SOCKET_NETWORK socket452  6531 driver7b
SOCKET_NETWORK socket453  6532 driver7c
SOCKET_NETWORK socket454  6533 driver7d
SOCKET_NETWORK socket455  6534 driver7e
SOCKET_NETWORK socket456  6535 driver7f
SOCKET_NETWORK socket457  6536 driver7a
SOCKET_NETWORK socket458  6537 driver7b
SOCKET_NETWORK socket459  6538 driver7c
SOCKET_NETWORK socket460  6539 driver7d
SOCKET_NETWORK socket461  6540 driver7e
SOCKET_NETWORK socket462  6541 driver7f
SOCKET_NETWORK socket463  6542 driver7a
SOCKET_NETWORK socket464  6543 driver7b
SOCKET_NETWORK socket465  6544 driver7c
SOCKET_NETWORK socket466  6545 driver7d
SOCKET_NETWORK socket467  6546 driver7e
SOCKET_NETWORK socket468  6547 driver7f
SOCKET_NETWORK socket469  6548 driver7a
SOCKET_NETWORK socket470  6549 driver7b
SOCKET_NETWORK socket471  6550 driver7c
SOCKET_NETWORK socket472  6551 driver7d
SOCKET_NETWORK socket473  6552 driver7e
SOCKET_NETWORK socket474  6553 driver7f
SOCKET_NETWORK socket475  6554 driver7a
SOCKET_NETWORK socket476  6555 driver7b
SOCKET_NETWORK socket477  6556 driver7c
SOCKET_NETWORK socket478  6557 driver7d
SOCKET_NETWORK socket479  6558 driver7e
SOCKET_NETWORK socket480  6559 driver7f
SOCKET_NETWORK socket481  6560 driver7a
SOCKET_NETWORK socket482  6561 driver7b
SOCKET_NETWORK socket483  6562 driver7c
SOCKET_NETWORK socket484  6563 driver7d
SOCKET_NETWORK socket485  6564 driver7e
SOCKET_NETWORK socket486  6565 driver7f
SOCKET_NETWORK socket487  6566 driver7a
SOCKET_NETWORK socket488  6567 driver7b
SOCKET_NETWORK socket489  6568 driver7c
SOCKET_NETWORK socket490  6569 driver7d
SOCKET_NETWORK socket491  6570 driver8a
SOCKET_NETWORK socket492  6571 driver8b
SOCKET_NETWORK socket493  6572 driver8c
SOCKET_NETWORK socket494  6573 driver8d
SOCKET_NETWORK socket495  6574 driver8e
SOCKET_NETWORK socket496  6575 driver8f
SOCKET_NETWORK socket497  6576 driver8a
SOCKET_NETWORK socket498  6577 driver8b
SOCKET_NETWORK socket499  6578 driver8c
SOCKET_NETWORK socket500  6579 driver8d
SOCKET_NETWORK socket501  6580 driver8e
SOCKET_NETWORK socket502  6581 driver8f
SOCKET_NETWORK socket503  6582 driver8a
SOCKET_NETWORK socket504  6583 driver8b
SOCKET_NETWORK socket505  6584 driver8c
SOCKET_NETWORK socket506  6585 driver8d
SOCKET_NETWORK socket507  6586 driver8e
SOCKET_NETWORK socket508  6587 driver8f
SOCKET_NETWORK socket509  6588 driver8a
SOCKET_NETWORK socket510  6589 driver8b
SOCKET_NETWORK socket511  6590 driver8c
SOCKET_NETWORK socket512  6591 driver8d
SOCKET_NETWORK socket513  6592 driver8e
SOCKET_NETWORK socket514  6593 driver8f
SOCKET_NETWORK socket515  6594 driver8a
SOCKET_NETWORK socket516  6595 driver8b
SOCKET_NETWORK socket517  6596 driver8c
SOCKET_NETWORK socket518  6597 driver8d
SOCKET_NETWORK socket519  6598 driver8e
SOCKET_NETWORK socket520  6599 driver8f
SOCKET_NETWORK socket521  6600 driver8a
SOCKET_NETWORK socket522  6601 driver8b
SOCKET_NETWORK socket523  6602 driver8c
SOCKET_NETWORK socket524  6603 driver8d
SOCKET_NETWORK socket525  6604 driver8e
SOCKET_NETWORK socket526  6605 driver8f
SOCKET_NETWORK socket527  6606 driver8a
SOCKET_NETWORK socket528  6607 driver8b
SOCKET_NETWORK socket529  6608 driver8c
SOCKET_NETWORK socket530  6609 driver8d
SOCKET_NETWORK socket531  6610 driver8e
SOCKET_NETWORK socket532  6611 driver8f
SOCKET_NETWORK socket533  6612 driver8a
SOCKET_NETWORK socket534  6613 driver8b
SOCKET_NETWORK socket535  6614 driver8c
SOCKET_NETWORK socket536  6615 driver8d
SOCKET_NETWORK socket537  6616 driver8e
SOCKET_NETWORK socket538  6617 driver8f
SOCKET_NETWORK socket539  6618 driver8a
SOCKET_NETWORK socket540  6619 driver8b
SOCKET_NETWORK socket541  6620 driver8c
SOCKET_NETWORK socket542  6621 driver8d
SOCKET_NETWORK socket543  6622 driver8e
SOCKET_NETWORK socket544  6623 driver8f
SOCKET_NETWORK socket545  6624 driver8a
SOCKET_NETWORK socket546  6625 driver8b
SOCKET_NETWORK socket547  6626 driver8c
SOCKET_NETWORK socket548  6627 driver8d
SOCKET_NETWORK socket549  6628 driver8e
SOCKET_NETWORK socket550  6629 driver8f
SOCKET_NETWORK socket551  6630 driver8a
SOCKET_NETWORK socket552  6631 driver8b
SOCKET_NETWORK socket553  6632 driver8c
SOCKET_NETWORK socket554  6633 driver8d
SOCKET_NETWORK socket555  6634 driver8e
SOCKET_NETWORK socket556  6635 driver8f
SOCKET_NETWORK socket557  6636 driver8a
SOCKET_NETWORK socket558  6637 driver8b
```

SOCKET_NETWORK socket559 6638 driver8c
SOCKET_NETWORK socket560 6639 driver8d
SOCKET_NETWORK socket561 6500 driver9a
SOCKET_NETWORK socket562 6501 driver9b
SOCKET_NETWORK socket563 6502 driver9c
SOCKET_NETWORK socket564 6503 driver9d
SOCKET_NETWORK socket565 6504 driver9e
SOCKET_NETWORK socket566 6505 driver9f
SOCKET_NETWORK socket567 6506 driver9a
SOCKET_NETWORK socket568 6507 driver9b
SOCKET_NETWORK socket569 6508 driver9c
SOCKET_NETWORK socket570 6509 driver9d
SOCKET_NETWORK socket571 6510 driver9e
SOCKET_NETWORK socket572 6511 driver9f
SOCKET_NETWORK socket573 6512 driver9a
SOCKET_NETWORK socket574 6513 driver9b
SOCKET_NETWORK socket575 6514 driver9c
SOCKET_NETWORK socket576 6515 driver9d
SOCKET_NETWORK socket577 6516 driver9e
SOCKET_NETWORK socket578 6517 driver9f
SOCKET_NETWORK socket579 6518 driver9a
SOCKET_NETWORK socket580 6519 driver9b
SOCKET_NETWORK socket581 6520 driver9c
SOCKET_NETWORK socket582 6521 driver9d
SOCKET_NETWORK socket583 6522 driver9e
SOCKET_NETWORK socket584 6523 driver9f
SOCKET_NETWORK socket585 6524 driver9a
SOCKET_NETWORK socket586 6525 driver9b
SOCKET_NETWORK socket587 6526 driver9c
SOCKET_NETWORK socket588 6527 driver9d
SOCKET_NETWORK socket589 6528 driver9e
SOCKET_NETWORK socket590 6529 driver9f
SOCKET_NETWORK socket591 6530 driver9a
SOCKET_NETWORK socket592 6531 driver9b
SOCKET_NETWORK socket593 6532 driver9c
SOCKET_NETWORK socket594 6533 driver9d
SOCKET_NETWORK socket595 6534 driver9e
SOCKET_NETWORK socket596 6535 driver9f
SOCKET_NETWORK socket597 6536 driver9a
SOCKET_NETWORK socket598 6537 driver9b
SOCKET_NETWORK socket599 6538 driver9c
SOCKET_NETWORK socket600 6539 driver9d
SOCKET_NETWORK socket601 6540 driver9e
SOCKET_NETWORK socket602 6541 driver9f
SOCKET_NETWORK socket603 6542 driver9a
SOCKET_NETWORK socket604 6543 driver9b
SOCKET_NETWORK socket605 6544 driver9c
SOCKET_NETWORK socket606 6545 driver9d
SOCKET_NETWORK socket607 6546 driver9e
SOCKET_NETWORK socket608 6547 driver9f
SOCKET_NETWORK socket609 6548 driver9a
SOCKET_NETWORK socket610 6549 driver9b
SOCKET_NETWORK socket611 6550 driver9c
SOCKET_NETWORK socket612 6551 driver9d
SOCKET_NETWORK socket613 6552 driver9e
SOCKET_NETWORK socket614 6553 driver9f
SOCKET_NETWORK socket615 6554 driver9a
SOCKET_NETWORK socket616 6555 driver9b
SOCKET_NETWORK socket617 6556 driver9c
SOCKET_NETWORK socket618 6557 driver9d
SOCKET_NETWORK socket619 6558 driver9e
SOCKET_NETWORK socket620 6559 driver9f
SOCKET_NETWORK socket621 6560 driver9a
SOCKET_NETWORK socket622 6561 driver9b
SOCKET_NETWORK socket623 6562 driver9c
SOCKET_NETWORK socket624 6563 driver9d
SOCKET_NETWORK socket625 6564 driver9e
SOCKET_NETWORK socket626 6565 driver9f
SOCKET_NETWORK socket627 6566 driver9a
SOCKET_NETWORK socket628 6567 driver9b
SOCKET_NETWORK socket629 6568 driver9c
SOCKET_NETWORK socket630 6569 driver9d
SOCKET_NETWORK socket631 6570 driver10a
SOCKET_NETWORK socket632 6571 driver10b
SOCKET_NETWORK socket633 6572 driver10c
SOCKET_NETWORK socket634 6573 driver10d
SOCKET_NETWORK socket635 6574 driver10e
SOCKET_NETWORK socket636 6575 driver10f
SOCKET_NETWORK socket637 6576 driver10a
SOCKET_NETWORK socket638 6577 driver10b
SOCKET_NETWORK socket639 6578 driver10c
SOCKET_NETWORK socket640 6579 driver10d
SOCKET_NETWORK socket641 6580 driver10e
SOCKET_NETWORK socket642 6581 driver10f
SOCKET_NETWORK socket643 6582 driver10a
SOCKET_NETWORK socket644 6583 driver10b
SOCKET_NETWORK socket645 6584 driver10c
SOCKET_NETWORK socket646 6585 driver10d
SOCKET_NETWORK socket647 6586 driver10e
SOCKET_NETWORK socket648 6587 driver10f
SOCKET_NETWORK socket649 6588 driver10a
SOCKET_NETWORK socket650 6589 driver10b
SOCKET_NETWORK socket651 6590 driver10c
SOCKET_NETWORK socket652 6591 driver10d
SOCKET_NETWORK socket653 6592 driver10e
SOCKET_NETWORK socket654 6593 driver10f
SOCKET_NETWORK socket655 6594 driver10a
SOCKET_NETWORK socket656 6595 driver10b
SOCKET_NETWORK socket657 6596 driver10c
SOCKET_NETWORK socket658 6597 driver10d
SOCKET_NETWORK socket659 6598 driver10e
SOCKET_NETWORK socket660 6599 driver10f
SOCKET_NETWORK socket661 6600 driver10a
SOCKET_NETWORK socket662 6601 driver10b
SOCKET_NETWORK socket663 6602 driver10c
SOCKET_NETWORK socket664 6603 driver10d
SOCKET_NETWORK socket665 6604 driver10e
SOCKET_NETWORK socket666 6605 driver10f
SOCKET_NETWORK socket667 6606 driver10a
SOCKET_NETWORK socket668 6607 driver10b
SOCKET_NETWORK socket669 6608 driver10c
SOCKET_NETWORK socket670 6609 driver10d
SOCKET_NETWORK socket671 6610 driver10e
SOCKET_NETWORK socket672 6611 driver10f
SOCKET_NETWORK socket673 6612 driver10a
SOCKET_NETWORK socket674 6613 driver10b
SOCKET_NETWORK socket675 6614 driver10c
SOCKET_NETWORK socket676 6615 driver10d
SOCKET_NETWORK socket677 6616 driver10e
SOCKET_NETWORK socket678 6617 driver10f
SOCKET_NETWORK socket679 6618 driver10a
SOCKET_NETWORK socket680 6619 driver10b
SOCKET_NETWORK socket681 6620 driver10c
SOCKET_NETWORK socket682 6621 driver10d
SOCKET_NETWORK socket683 6622 driver10e
SOCKET_NETWORK socket684 6623 driver10f
SOCKET_NETWORK socket685 6624 driver10a
SOCKET_NETWORK socket686 6625 driver10b
SOCKET_NETWORK socket687 6626 driver10c
SOCKET_NETWORK socket688 6627 driver10d
SOCKET_NETWORK socket689 6628 driver10e
SOCKET_NETWORK socket690 6629 driver10f
SOCKET_NETWORK socket691 6630 driver10a
SOCKET_NETWORK socket692 6631 driver10b
SOCKET_NETWORK socket693 6632 driver10c
SOCKET_NETWORK socket694 6633 driver10d
SOCKET_NETWORK socket695 6634 driver10e
SOCKET_NETWORK socket696 6635 driver10f
SOCKET_NETWORK socket697 6636 driver10a
SOCKET_NETWORK socket698 6637 driver10b
SOCKET_NETWORK socket699 6638 driver10c
SOCKET_NETWORK socket700 6639 driver10d
SOCKET_NETWORK socket701 6500 driver11a
SOCKET_NETWORK socket702 6501 driver11b
SOCKET_NETWORK socket703 6502 driver11c
SOCKET_NETWORK socket704 6503 driver11d
SOCKET_NETWORK socket705 6504 driver11e
SOCKET_NETWORK socket706 6505 driver11f
SOCKET_NETWORK socket707 6506 driver11a
SOCKET_NETWORK socket708 6507 driver11b
SOCKET_NETWORK socket709 6508 driver11c
SOCKET_NETWORK socket710 6509 driver11d
SOCKET_NETWORK socket711 6510 driver11e
SOCKET_NETWORK socket712 6511 driver11f
SOCKET_NETWORK socket713 6512 driver11a
SOCKET_NETWORK socket714 6513 driver11b
SOCKET_NETWORK socket715 6514 driver11c
SOCKET_NETWORK socket716 6515 driver11d
SOCKET_NETWORK socket717 6516 driver11e
SOCKET_NETWORK socket718 6517 driver11f
SOCKET_NETWORK socket719 6518 driver11a
SOCKET_NETWORK socket720 6519 driver11b
SOCKET_NETWORK socket721 6520 driver11c
SOCKET_NETWORK socket722 6521 driver11d
SOCKET_NETWORK socket723 6522 driver11e
SOCKET_NETWORK socket724 6523 driver11f
SOCKET_NETWORK socket725 6524 driver11a
SOCKET_NETWORK socket726 6525 driver11b
SOCKET_NETWORK socket727 6526 driver11c
SOCKET_NETWORK socket728 6527 driver11d
SOCKET_NETWORK socket729 6528 driver11e
SOCKET_NETWORK socket730 6529 driver11f
SOCKET_NETWORK socket731 6530 driver11a
SOCKET_NETWORK socket732 6531 driver11b
SOCKET_NETWORK socket733 6532 driver11c
SOCKET_NETWORK socket734 6533 driver11d
SOCKET_NETWORK socket735 6534 driver11e
SOCKET_NETWORK socket736 6535 driver11f
SOCKET_NETWORK socket737 6536 driver11a
SOCKET_NETWORK socket738 6537 driver11b
SOCKET_NETWORK socket739 6538 driver11c
SOCKET_NETWORK socket740 6539 driver11d
SOCKET_NETWORK socket741 6540 driver11e
SOCKET_NETWORK socket742 6541 driver11f
SOCKET_NETWORK socket743 6542 driver11a
SOCKET_NETWORK socket744 6543 driver11b
SOCKET_NETWORK socket745 6544 driver11c
SOCKET_NETWORK socket746 6545 driver11d
SOCKET_NETWORK socket747 6546 driver11e
SOCKET_NETWORK socket748 6547 driver11f
SOCKET_NETWORK socket749 6548 driver11a
SOCKET_NETWORK socket750 6549 driver11b
SOCKET_NETWORK socket751 6550 driver11c
SOCKET_NETWORK socket752 6551 driver11d
SOCKET_NETWORK socket753 6552 driver11e
SOCKET_NETWORK socket754 6553 driver11f
SOCKET_NETWORK socket755 6554 driver11a
SOCKET_NETWORK socket756 6555 driver11b

```
SOCKET_NETWORK socket757  6556 driver11c
SOCKET_NETWORK socket758  6557 driver11d
SOCKET_NETWORK socket759  6558 driver11e
SOCKET_NETWORK socket760  6559 driver11f
SOCKET_NETWORK socket761  6560 driver11a
SOCKET_NETWORK socket762  6561 driver11b
SOCKET_NETWORK socket763  6562 driver11c
SOCKET_NETWORK socket764  6563 driver11d
SOCKET_NETWORK socket765  6564 driver11e
SOCKET_NETWORK socket766  6565 driver11f
SOCKET_NETWORK socket767  6566 driver11a
SOCKET_NETWORK socket768  6567 driver11b
SOCKET_NETWORK socket769  6568 driver11c
SOCKET_NETWORK socket770  6569 driver11d
SOCKET_NETWORK socket771  6570 driver12a
SOCKET_NETWORK socket772  6571 driver12b
SOCKET_NETWORK socket773  6572 driver12c
SOCKET_NETWORK socket774  6573 driver12d
SOCKET_NETWORK socket775  6574 driver12e
SOCKET_NETWORK socket776  6575 driver12f
SOCKET_NETWORK socket777  6576 driver12a
SOCKET_NETWORK socket778  6577 driver12b
SOCKET_NETWORK socket779  6578 driver12c
SOCKET_NETWORK socket780  6579 driver12d
SOCKET_NETWORK socket781  6580 driver12e
SOCKET_NETWORK socket782  6581 driver12f
SOCKET_NETWORK socket783  6582 driver12a
SOCKET_NETWORK socket784  6583 driver12b
SOCKET_NETWORK socket785  6584 driver12c
SOCKET_NETWORK socket786  6585 driver12d
SOCKET_NETWORK socket787  6586 driver12e
SOCKET_NETWORK socket788  6587 driver12f
SOCKET_NETWORK socket789  6588 driver12a
SOCKET_NETWORK socket790  6589 driver12b
SOCKET_NETWORK socket791  6590 driver12c
SOCKET_NETWORK socket792  6591 driver12d
SOCKET_NETWORK socket793  6592 driver12e
SOCKET_NETWORK socket794  6593 driver12f
SOCKET_NETWORK socket795  6594 driver12a
SOCKET_NETWORK socket796  6595 driver12b
SOCKET_NETWORK socket797  6596 driver12c
SOCKET_NETWORK socket798  6597 driver12d
SOCKET_NETWORK socket799  6598 driver12e
SOCKET_NETWORK socket800  6599 driver12f
SOCKET_NETWORK socket801  6600 driver12a
SOCKET_NETWORK socket802  6601 driver12b
SOCKET_NETWORK socket803  6602 driver12c
SOCKET_NETWORK socket804  6603 driver12d
SOCKET_NETWORK socket805  6604 driver12e
SOCKET_NETWORK socket806  6605 driver12f
SOCKET_NETWORK socket807  6606 driver12a
SOCKET_NETWORK socket808  6607 driver12b
SOCKET_NETWORK socket809  6608 driver12c
SOCKET_NETWORK socket810  6609 driver12d
SOCKET_NETWORK socket811  6610 driver12e
SOCKET_NETWORK socket812  6611 driver12f
SOCKET_NETWORK socket813  6612 driver12a
SOCKET_NETWORK socket814  6613 driver12b
SOCKET_NETWORK socket815  6614 driver12c
SOCKET_NETWORK socket816  6615 driver12d
SOCKET_NETWORK socket817  6616 driver12e
SOCKET_NETWORK socket818  6617 driver12f
SOCKET_NETWORK socket819  6618 driver12a
SOCKET_NETWORK socket820  6619 driver12b
SOCKET_NETWORK socket821  6620 driver12c
SOCKET_NETWORK socket822  6621 driver12d
SOCKET_NETWORK socket823  6622 driver12e
SOCKET_NETWORK socket824  6623 driver12f
SOCKET_NETWORK socket825  6624 driver12a
SOCKET_NETWORK socket826  6625 driver12b
SOCKET_NETWORK socket827  6626 driver12c
SOCKET_NETWORK socket828  6627 driver12d
SOCKET_NETWORK socket829  6628 driver12e
SOCKET_NETWORK socket830  6629 driver12f
SOCKET_NETWORK socket831  6630 driver12a
SOCKET_NETWORK socket832  6631 driver12b
SOCKET_NETWORK socket833  6632 driver12c
SOCKET_NETWORK socket834  6633 driver12d
SOCKET_NETWORK socket835  6634 driver12e
SOCKET_NETWORK socket836  6635 driver12f
SOCKET_NETWORK socket837  6636 driver12a
SOCKET_NETWORK socket838  6637 driver12b
SOCKET_NETWORK socket839  6638 driver12c
SOCKET_NETWORK socket840  6639 driver12d
#elif MASTER_NUM3
SOCKET_NETWORK socket841  6500 driver13a
SOCKET_NETWORK socket842  6501 driver13b
SOCKET_NETWORK socket843  6502 driver13c
SOCKET_NETWORK socket844  6503 driver13d
SOCKET_NETWORK socket845  6504 driver13e
SOCKET_NETWORK socket846  6505 driver13f
SOCKET_NETWORK socket847  6506 driver13a
SOCKET_NETWORK socket848  6507 driver13b
SOCKET_NETWORK socket849  6508 driver13c
SOCKET_NETWORK socket850  6509 driver13d
SOCKET_NETWORK socket851  6510 driver13e
SOCKET_NETWORK socket852  6511 driver13f
SOCKET_NETWORK socket853  6512 driver13a
SOCKET_NETWORK socket854  6513 driver13b
SOCKET_NETWORK socket855  6514 driver13c
SOCKET_NETWORK socket856  6515 driver13d
SOCKET_NETWORK socket857  6516 driver13e
SOCKET_NETWORK socket858  6517 driver13f
SOCKET_NETWORK socket859  6518 driver13a
SOCKET_NETWORK socket860  6519 driver13b
SOCKET_NETWORK socket861  6520 driver13c
SOCKET_NETWORK socket862  6521 driver13d
SOCKET_NETWORK socket863  6522 driver13e
SOCKET_NETWORK socket864  6523 driver13f
SOCKET_NETWORK socket865  6524 driver13a
SOCKET_NETWORK socket866  6525 driver13b
SOCKET_NETWORK socket867  6526 driver13c
SOCKET_NETWORK socket868  6527 driver13d
SOCKET_NETWORK socket869  6528 driver13e
SOCKET_NETWORK socket870  6529 driver13f
SOCKET_NETWORK socket871  6530 driver13a
SOCKET_NETWORK socket872  6531 driver13b
SOCKET_NETWORK socket873  6532 driver13c
SOCKET_NETWORK socket874  6533 driver13d
SOCKET_NETWORK socket875  6534 driver13e
SOCKET_NETWORK socket876  6535 driver13f
SOCKET_NETWORK socket877  6536 driver13a
SOCKET_NETWORK socket878  6537 driver13b
SOCKET_NETWORK socket879  6538 driver13c
SOCKET_NETWORK socket880  6539 driver13d
SOCKET_NETWORK socket881  6540 driver13e
SOCKET_NETWORK socket882  6541 driver13f
SOCKET_NETWORK socket883  6542 driver13a
SOCKET_NETWORK socket884  6543 driver13b
SOCKET_NETWORK socket885  6544 driver13c
SOCKET_NETWORK socket886  6545 driver13d
SOCKET_NETWORK socket887  6546 driver13e
SOCKET_NETWORK socket888  6547 driver13f
SOCKET_NETWORK socket889  6548 driver13a
SOCKET_NETWORK socket890  6549 driver13b
SOCKET_NETWORK socket891  6550 driver13c
SOCKET_NETWORK socket892  6551 driver13d
SOCKET_NETWORK socket893  6552 driver13e
SOCKET_NETWORK socket894  6553 driver13f
SOCKET_NETWORK socket895  6554 driver13a
SOCKET_NETWORK socket896  6555 driver13b
SOCKET_NETWORK socket897  6556 driver13c
SOCKET_NETWORK socket898  6557 driver13d
SOCKET_NETWORK socket899  6558 driver13e
SOCKET_NETWORK socket900  6559 driver13f
SOCKET_NETWORK socket901  6560 driver13a
SOCKET_NETWORK socket902  6561 driver13b
SOCKET_NETWORK socket903  6562 driver13c
SOCKET_NETWORK socket904  6563 driver13d
SOCKET_NETWORK socket905  6564 driver13e
SOCKET_NETWORK socket906  6565 driver13f
SOCKET_NETWORK socket907  6566 driver13a
SOCKET_NETWORK socket908  6567 driver13b
SOCKET_NETWORK socket909  6568 driver13c
SOCKET_NETWORK socket910  6569 driver13d
SOCKET_NETWORK socket911  6570 driver14a
SOCKET_NETWORK socket912  6571 driver14b
SOCKET_NETWORK socket913  6572 driver14c
SOCKET_NETWORK socket914  6573 driver14d
SOCKET_NETWORK socket915  6574 driver14e
SOCKET_NETWORK socket916  6575 driver14f
SOCKET_NETWORK socket917  6576 driver14a
SOCKET_NETWORK socket918  6577 driver14b
SOCKET_NETWORK socket919  6578 driver14c
SOCKET_NETWORK socket920  6579 driver14d
SOCKET_NETWORK socket921  6580 driver14e
SOCKET_NETWORK socket922  6581 driver14f
SOCKET_NETWORK socket923  6582 driver14a
SOCKET_NETWORK socket924  6583 driver14b
SOCKET_NETWORK socket925  6584 driver14c
SOCKET_NETWORK socket926  6585 driver14d
SOCKET_NETWORK socket927  6586 driver14e
SOCKET_NETWORK socket928  6587 driver14f
SOCKET_NETWORK socket929  6588 driver14a
SOCKET_NETWORK socket930  6589 driver14b
SOCKET_NETWORK socket931  6590 driver14c
SOCKET_NETWORK socket932  6591 driver14d
SOCKET_NETWORK socket933  6592 driver14e
SOCKET_NETWORK socket934  6593 driver14f
SOCKET_NETWORK socket935  6594 driver14a
SOCKET_NETWORK socket936  6595 driver14b
SOCKET_NETWORK socket937  6596 driver14c
SOCKET_NETWORK socket938  6597 driver14d
SOCKET_NETWORK socket939  6598 driver14e
SOCKET_NETWORK socket940  6599 driver14f
SOCKET_NETWORK socket941  6600 driver14a
SOCKET_NETWORK socket942  6601 driver14b
SOCKET_NETWORK socket943  6602 driver14c
SOCKET_NETWORK socket944  6603 driver14d
SOCKET_NETWORK socket945  6604 driver14e
SOCKET_NETWORK socket946  6605 driver14f
SOCKET_NETWORK socket947  6606 driver14a
SOCKET_NETWORK socket948  6607 driver14b
SOCKET_NETWORK socket949  6608 driver14c
SOCKET_NETWORK socket950  6609 driver14d
SOCKET_NETWORK socket951  6610 driver14e
SOCKET_NETWORK socket952  6611 driver14f
SOCKET_NETWORK socket953  6612 driver14a
```

SOCKET_NETWORK socket954 6613 driver14b
SOCKET_NETWORK socket955 6614 driver14c
SOCKET_NETWORK socket956 6615 driver14d
SOCKET_NETWORK socket957 6616 driver14e
SOCKET_NETWORK socket958 6617 driver14f
SOCKET_NETWORK socket959 6618 driver14a
SOCKET_NETWORK socket960 6619 driver14b
SOCKET_NETWORK socket961 6620 driver14c
SOCKET_NETWORK socket962 6621 driver14d
SOCKET_NETWORK socket963 6622 driver14e
SOCKET_NETWORK socket964 6623 driver14f
SOCKET_NETWORK socket965 6624 driver14a
SOCKET_NETWORK socket966 6625 driver14b
SOCKET_NETWORK socket967 6626 driver14c
SOCKET_NETWORK socket968 6627 driver14d
SOCKET_NETWORK socket969 6628 driver14e
SOCKET_NETWORK socket970 6629 driver14f
SOCKET_NETWORK socket971 6630 driver14a
SOCKET_NETWORK socket972 6631 driver14b
SOCKET_NETWORK socket973 6632 driver14c
SOCKET_NETWORK socket974 6633 driver14d
SOCKET_NETWORK socket975 6634 driver14e
SOCKET_NETWORK socket976 6635 driver14f
SOCKET_NETWORK socket977 6636 driver14a
SOCKET_NETWORK socket978 6637 driver14b
SOCKET_NETWORK socket979 6638 driver14c
SOCKET_NETWORK socket980 6639 driver14d
SOCKET_NETWORK socket981 6500 driver15a
SOCKET_NETWORK socket982 6501 driver15b
SOCKET_NETWORK socket983 6502 driver15c
SOCKET_NETWORK socket984 6503 driver15d
SOCKET_NETWORK socket985 6504 driver15e
SOCKET_NETWORK socket986 6505 driver15f
SOCKET_NETWORK socket987 6506 driver15a
SOCKET_NETWORK socket988 6507 driver15b
SOCKET_NETWORK socket989 6508 driver15c
SOCKET_NETWORK socket990 6509 driver15d
SOCKET_NETWORK socket991 6510 driver15e
SOCKET_NETWORK socket992 6511 driver15f
SOCKET_NETWORK socket993 6512 driver15a
SOCKET_NETWORK socket994 6513 driver15b
SOCKET_NETWORK socket995 6514 driver15c
SOCKET_NETWORK socket996 6515 driver15d
SOCKET_NETWORK socket997 6516 driver15e
SOCKET_NETWORK socket998 6517 driver15f
SOCKET_NETWORK socket999 6518 driver15a
SOCKET_NETWORK socket1000 6519 driver15b
SOCKET_NETWORK socket1001 6520 driver15c
SOCKET_NETWORK socket1002 6521 driver15d
SOCKET_NETWORK socket1003 6522 driver15e
SOCKET_NETWORK socket1004 6523 driver15f
SOCKET_NETWORK socket1005 6524 driver15a
SOCKET_NETWORK socket1006 6525 driver15b
SOCKET_NETWORK socket1007 6526 driver15c
SOCKET_NETWORK socket1008 6527 driver15d
SOCKET_NETWORK socket1009 6528 driver15e
SOCKET_NETWORK socket1010 6529 driver15f
SOCKET_NETWORK socket1011 6530 driver15a
SOCKET_NETWORK socket1012 6531 driver15b
SOCKET_NETWORK socket1013 6532 driver15c
SOCKET_NETWORK socket1014 6533 driver15d
SOCKET_NETWORK socket1015 6534 driver15e
SOCKET_NETWORK socket1016 6535 driver15f
SOCKET_NETWORK socket1017 6536 driver15a
SOCKET_NETWORK socket1018 6537 driver15b
SOCKET_NETWORK socket1019 6538 driver15c
SOCKET_NETWORK socket1020 6539 driver15d
SOCKET_NETWORK socket1021 6540 driver15e
SOCKET_NETWORK socket1022 6541 driver15f
SOCKET_NETWORK socket1023 6542 driver15a
SOCKET_NETWORK socket1024 6543 driver15b
SOCKET_NETWORK socket1025 6544 driver15c
SOCKET_NETWORK socket1026 6545 driver15d
SOCKET_NETWORK socket1027 6546 driver15e
SOCKET_NETWORK socket1028 6547 driver15f
SOCKET_NETWORK socket1029 6548 driver15a
SOCKET_NETWORK socket1030 6549 driver15b
SOCKET_NETWORK socket1031 6550 driver15c
SOCKET_NETWORK socket1032 6551 driver15d
SOCKET_NETWORK socket1033 6552 driver15e
SOCKET_NETWORK socket1034 6553 driver15f
SOCKET_NETWORK socket1035 6554 driver15a
SOCKET_NETWORK socket1036 6555 driver15b
SOCKET_NETWORK socket1037 6556 driver15c
SOCKET_NETWORK socket1038 6557 driver15d
SOCKET_NETWORK socket1039 6558 driver15e
SOCKET_NETWORK socket1040 6559 driver15f
SOCKET_NETWORK socket1041 6560 driver15a
SOCKET_NETWORK socket1042 6561 driver15b
SOCKET_NETWORK socket1043 6562 driver15c
SOCKET_NETWORK socket1044 6563 driver15d
SOCKET_NETWORK socket1045 6564 driver15e
SOCKET_NETWORK socket1046 6565 driver15f
SOCKET_NETWORK socket1047 6566 driver15a
SOCKET_NETWORK socket1048 6567 driver15b
SOCKET_NETWORK socket1049 6568 driver15c
SOCKET_NETWORK socket1050 6569 driver15d
SOCKET_NETWORK socket1051 6570 driver16a
SOCKET_NETWORK socket1052 6571 driver16b
SOCKET_NETWORK socket1053 6572 driver16c
SOCKET_NETWORK socket1054 6573 driver16d
SOCKET_NETWORK socket1055 6574 driver16e
SOCKET_NETWORK socket1056 6575 driver16f
SOCKET_NETWORK socket1057 6576 driver16a
SOCKET_NETWORK socket1058 6577 driver16b
SOCKET_NETWORK socket1059 6578 driver16c
SOCKET_NETWORK socket1060 6579 driver16d
SOCKET_NETWORK socket1061 6580 driver16e
SOCKET_NETWORK socket1062 6581 driver16f
SOCKET_NETWORK socket1063 6582 driver16a
SOCKET_NETWORK socket1064 6583 driver16b
SOCKET_NETWORK socket1065 6584 driver16c
SOCKET_NETWORK socket1066 6585 driver16d
SOCKET_NETWORK socket1067 6586 driver16e
SOCKET_NETWORK socket1068 6587 driver16f
SOCKET_NETWORK socket1069 6588 driver16a
SOCKET_NETWORK socket1070 6589 driver16b
SOCKET_NETWORK socket1071 6590 driver16c
SOCKET_NETWORK socket1072 6591 driver16d
SOCKET_NETWORK socket1073 6592 driver16e
SOCKET_NETWORK socket1074 6593 driver16f
SOCKET_NETWORK socket1075 6594 driver16a
SOCKET_NETWORK socket1076 6595 driver16b
SOCKET_NETWORK socket1077 6596 driver16c
SOCKET_NETWORK socket1078 6597 driver16d
SOCKET_NETWORK socket1079 6598 driver16e
SOCKET_NETWORK socket1080 6599 driver16f
SOCKET_NETWORK socket1081 6600 driver16a
SOCKET_NETWORK socket1082 6601 driver16b
SOCKET_NETWORK socket1083 6602 driver16c
SOCKET_NETWORK socket1084 6603 driver16d
SOCKET_NETWORK socket1085 6604 driver16e
SOCKET_NETWORK socket1086 6605 driver16f
SOCKET_NETWORK socket1087 6606 driver16a
SOCKET_NETWORK socket1088 6607 driver16b
SOCKET_NETWORK socket1089 6608 driver16c
SOCKET_NETWORK socket1090 6609 driver16d
SOCKET_NETWORK socket1091 6610 driver16e
SOCKET_NETWORK socket1092 6611 driver16f
SOCKET_NETWORK socket1093 6612 driver16a
SOCKET_NETWORK socket1094 6613 driver16b
SOCKET_NETWORK socket1095 6614 driver16c
SOCKET_NETWORK socket1096 6615 driver16d
SOCKET_NETWORK socket1097 6616 driver16e
SOCKET_NETWORK socket1098 6617 driver16f
SOCKET_NETWORK socket1099 6618 driver16a
SOCKET_NETWORK socket1100 6619 driver16b
SOCKET_NETWORK socket1101 6620 driver16c
SOCKET_NETWORK socket1102 6621 driver16d
SOCKET_NETWORK socket1103 6622 driver16e
SOCKET_NETWORK socket1104 6623 driver16f
SOCKET_NETWORK socket1105 6624 driver16a
SOCKET_NETWORK socket1106 6625 driver16b
SOCKET_NETWORK socket1107 6626 driver16c
SOCKET_NETWORK socket1108 6627 driver16d
SOCKET_NETWORK socket1109 6628 driver16e
SOCKET_NETWORK socket1110 6629 driver16f
SOCKET_NETWORK socket1111 6630 driver16a
SOCKET_NETWORK socket1112 6631 driver16b
SOCKET_NETWORK socket1113 6632 driver16c
SOCKET_NETWORK socket1114 6633 driver16d
SOCKET_NETWORK socket1115 6634 driver16e
SOCKET_NETWORK socket1116 6635 driver16f
SOCKET_NETWORK socket1117 6636 driver16a
SOCKET_NETWORK socket1118 6637 driver16b
SOCKET_NETWORK socket1119 6638 driver16c
SOCKET_NETWORK socket1120 6639 driver16d
SOCKET_NETWORK socket1121 6500 driver17a
SOCKET_NETWORK socket1122 6501 driver17b
SOCKET_NETWORK socket1123 6502 driver17c
SOCKET_NETWORK socket1124 6503 driver17d
SOCKET_NETWORK socket1125 6504 driver17e
SOCKET_NETWORK socket1126 6505 driver17f
SOCKET_NETWORK socket1127 6506 driver17a
SOCKET_NETWORK socket1128 6507 driver17b
SOCKET_NETWORK socket1129 6508 driver17c
SOCKET_NETWORK socket1130 6509 driver17d
SOCKET_NETWORK socket1131 6510 driver17e
SOCKET_NETWORK socket1132 6511 driver17f
SOCKET_NETWORK socket1133 6512 driver17a
SOCKET_NETWORK socket1134 6513 driver17b
SOCKET_NETWORK socket1135 6514 driver17c
SOCKET_NETWORK socket1136 6515 driver17d
SOCKET_NETWORK socket1137 6516 driver17e
SOCKET_NETWORK socket1138 6517 driver17f
SOCKET_NETWORK socket1139 6518 driver17a
SOCKET_NETWORK socket1140 6519 driver17b
SOCKET_NETWORK socket1141 6520 driver17c
SOCKET_NETWORK socket1142 6521 driver17d
SOCKET_NETWORK socket1143 6522 driver17e
SOCKET_NETWORK socket1144 6523 driver17f
SOCKET_NETWORK socket1145 6524 driver17a
SOCKET_NETWORK socket1146 6525 driver17b
SOCKET_NETWORK socket1147 6526 driver17c
SOCKET_NETWORK socket1148 6527 driver17d
SOCKET_NETWORK socket1149 6528 driver17e
SOCKET_NETWORK socket1150 6529 driver17f
SOCKET_NETWORK socket1151 6530 driver17a

```
SOCKET_NETWORK socket1152  6531 driver17b
SOCKET_NETWORK socket1153  6532 driver17c
SOCKET_NETWORK socket1154  6533 driver17d
SOCKET_NETWORK socket1155  6534 driver17e
SOCKET_NETWORK socket1156  6535 driver17f
SOCKET_NETWORK socket1157  6536 driver17a
SOCKET_NETWORK socket1158  6537 driver17b
SOCKET_NETWORK socket1159  6538 driver17c
SOCKET_NETWORK socket1160  6539 driver17d
SOCKET_NETWORK socket1161  6540 driver17e
SOCKET_NETWORK socket1162  6541 driver17f
SOCKET_NETWORK socket1163  6542 driver17a
SOCKET_NETWORK socket1164  6543 driver17b
SOCKET_NETWORK socket1165  6544 driver17c
SOCKET_NETWORK socket1166  6545 driver17d
SOCKET_NETWORK socket1167  6546 driver17e
SOCKET_NETWORK socket1168  6547 driver17f
SOCKET_NETWORK socket1169  6548 driver17a
SOCKET_NETWORK socket1170  6549 driver17b
SOCKET_NETWORK socket1171  6550 driver17c
SOCKET_NETWORK socket1172  6551 driver17d
SOCKET_NETWORK socket1173  6552 driver17e
SOCKET_NETWORK socket1174  6553 driver17f
SOCKET_NETWORK socket1175  6554 driver17a
SOCKET_NETWORK socket1176  6555 driver17b
SOCKET_NETWORK socket1177  6556 driver17c
SOCKET_NETWORK socket1178  6557 driver17d
SOCKET_NETWORK socket1179  6558 driver17e
SOCKET_NETWORK socket1180  6559 driver17f
SOCKET_NETWORK socket1181  6560 driver17a
SOCKET_NETWORK socket1182  6561 driver17b
SOCKET_NETWORK socket1183  6562 driver17c
SOCKET_NETWORK socket1184  6563 driver17d
SOCKET_NETWORK socket1185  6564 driver17e
SOCKET_NETWORK socket1186  6565 driver17f
SOCKET_NETWORK socket1187  6566 driver17a
SOCKET_NETWORK socket1188  6567 driver17b
SOCKET_NETWORK socket1189  6568 driver17c
SOCKET_NETWORK socket1190  6569 driver17d
SOCKET_NETWORK socket1191  6570 driver18a
SOCKET_NETWORK socket1192  6571 driver18b
SOCKET_NETWORK socket1193  6572 driver18c
SOCKET_NETWORK socket1194  6573 driver18d
SOCKET_NETWORK socket1195  6574 driver18e
SOCKET_NETWORK socket1196  6575 driver18f
SOCKET_NETWORK socket1197  6576 driver18a
SOCKET_NETWORK socket1198  6577 driver18b
SOCKET_NETWORK socket1199  6578 driver18c
SOCKET_NETWORK socket1200  6579 driver18d
SOCKET_NETWORK socket1201  6580 driver18e
SOCKET_NETWORK socket1202  6581 driver18f
SOCKET_NETWORK socket1203  6582 driver18a
SOCKET_NETWORK socket1204  6583 driver18b
SOCKET_NETWORK socket1205  6584 driver18c
SOCKET_NETWORK socket1206  6585 driver18d
SOCKET_NETWORK socket1207  6586 driver18e
SOCKET_NETWORK socket1208  6587 driver18f
SOCKET_NETWORK socket1209  6588 driver18a
SOCKET_NETWORK socket1210  6589 driver18b
SOCKET_NETWORK socket1211  6590 driver18c
SOCKET_NETWORK socket1212  6591 driver18d
SOCKET_NETWORK socket1213  6592 driver18e
SOCKET_NETWORK socket1214  6593 driver18f
SOCKET_NETWORK socket1215  6594 driver18a
SOCKET_NETWORK socket1216  6595 driver18b
SOCKET_NETWORK socket1217  6596 driver18c
SOCKET_NETWORK socket1218  6597 driver18d
SOCKET_NETWORK socket1219  6598 driver18e
SOCKET_NETWORK socket1220  6599 driver18f
SOCKET_NETWORK socket1221  6600 driver18a
SOCKET_NETWORK socket1222  6601 driver18b
SOCKET_NETWORK socket1223  6602 driver18c
SOCKET_NETWORK socket1224  6603 driver18d
SOCKET_NETWORK socket1225  6604 driver18e
SOCKET_NETWORK socket1226  6605 driver18f
SOCKET_NETWORK socket1227  6606 driver18a
SOCKET_NETWORK socket1228  6607 driver18b
SOCKET_NETWORK socket1229  6608 driver18c
SOCKET_NETWORK socket1230  6609 driver18d
SOCKET_NETWORK socket1231  6610 driver18e
SOCKET_NETWORK socket1232  6611 driver18f
SOCKET_NETWORK socket1233  6612 driver18a
SOCKET_NETWORK socket1234  6613 driver18b
SOCKET_NETWORK socket1235  6614 driver18c
SOCKET_NETWORK socket1236  6615 driver18d
SOCKET_NETWORK socket1237  6616 driver18e
SOCKET_NETWORK socket1238  6617 driver18f
SOCKET_NETWORK socket1239  6618 driver18a
SOCKET_NETWORK socket1240  6619 driver18b
SOCKET_NETWORK socket1241  6620 driver18c
SOCKET_NETWORK socket1242  6621 driver18d
SOCKET_NETWORK socket1243  6622 driver18e
SOCKET_NETWORK socket1244  6623 driver18f
SOCKET_NETWORK socket1245  6624 driver18a
SOCKET_NETWORK socket1246  6625 driver18b
SOCKET_NETWORK socket1247  6626 driver18c
SOCKET_NETWORK socket1248  6627 driver18d
SOCKET_NETWORK socket1249  6628 driver18e
SOCKET_NETWORK socket1250  6629 driver18f
SOCKET_NETWORK socket1251  6630 driver18a
SOCKET_NETWORK socket1252  6631 driver18b
SOCKET_NETWORK socket1253  6632 driver18c
SOCKET_NETWORK socket1254  6633 driver18d
SOCKET_NETWORK socket1255  6634 driver18e
SOCKET_NETWORK socket1256  6635 driver18f
SOCKET_NETWORK socket1257  6636 driver18a
SOCKET_NETWORK socket1258  6637 driver18b
SOCKET_NETWORK socket1259  6638 driver18c
SOCKET_NETWORK socket1260  6639 driver18d
#elif MASTER_NUM4
SOCKET_NETWORK socket1261  6500 driver19a
SOCKET_NETWORK socket1262  6501 driver19b
SOCKET_NETWORK socket1263  6502 driver19c
SOCKET_NETWORK socket1264  6503 driver19d
SOCKET_NETWORK socket1265  6504 driver19e
SOCKET_NETWORK socket1266  6505 driver19f
SOCKET_NETWORK socket1267  6506 driver19a
SOCKET_NETWORK socket1268  6507 driver19b
SOCKET_NETWORK socket1269  6508 driver19c
SOCKET_NETWORK socket1270  6509 driver19d
SOCKET_NETWORK socket1271  6510 driver19e
SOCKET_NETWORK socket1272  6511 driver19f
SOCKET_NETWORK socket1273  6512 driver19a
SOCKET_NETWORK socket1274  6513 driver19b
SOCKET_NETWORK socket1275  6514 driver19c
SOCKET_NETWORK socket1276  6515 driver19d
SOCKET_NETWORK socket1277  6516 driver19e
SOCKET_NETWORK socket1278  6517 driver19f
SOCKET_NETWORK socket1279  6518 driver19a
SOCKET_NETWORK socket1280  6519 driver19b
SOCKET_NETWORK socket1281  6520 driver19c
SOCKET_NETWORK socket1282  6521 driver19d
SOCKET_NETWORK socket1283  6522 driver19e
SOCKET_NETWORK socket1284  6523 driver19f
SOCKET_NETWORK socket1285  6524 driver19a
SOCKET_NETWORK socket1286  6525 driver19b
SOCKET_NETWORK socket1287  6526 driver19c
SOCKET_NETWORK socket1288  6527 driver19d
SOCKET_NETWORK socket1289  6528 driver19e
SOCKET_NETWORK socket1290  6529 driver19f
SOCKET_NETWORK socket1291  6530 driver19a
SOCKET_NETWORK socket1292  6531 driver19b
SOCKET_NETWORK socket1293  6532 driver19c
SOCKET_NETWORK socket1294  6533 driver19d
SOCKET_NETWORK socket1295  6534 driver19e
SOCKET_NETWORK socket1296  6535 driver19f
SOCKET_NETWORK socket1297  6536 driver19a
SOCKET_NETWORK socket1298  6537 driver19b
SOCKET_NETWORK socket1299  6538 driver19c
SOCKET_NETWORK socket1300  6539 driver19d
SOCKET_NETWORK socket1301  6540 driver19e
SOCKET_NETWORK socket1302  6541 driver19f
SOCKET_NETWORK socket1303  6542 driver19a
SOCKET_NETWORK socket1304  6543 driver19b
SOCKET_NETWORK socket1305  6544 driver19c
SOCKET_NETWORK socket1306  6545 driver19d
SOCKET_NETWORK socket1307  6546 driver19e
SOCKET_NETWORK socket1308  6547 driver19f
SOCKET_NETWORK socket1309  6548 driver19a
SOCKET_NETWORK socket1310  6549 driver19b
SOCKET_NETWORK socket1311  6550 driver19c
SOCKET_NETWORK socket1312  6551 driver19d
SOCKET_NETWORK socket1313  6552 driver19e
SOCKET_NETWORK socket1314  6553 driver19f
SOCKET_NETWORK socket1315  6554 driver19a
SOCKET_NETWORK socket1316  6555 driver19b
SOCKET_NETWORK socket1317  6556 driver19c
SOCKET_NETWORK socket1318  6557 driver19d
SOCKET_NETWORK socket1319  6558 driver19e
SOCKET_NETWORK socket1320  6559 driver19f
SOCKET_NETWORK socket1321  6560 driver19a
SOCKET_NETWORK socket1322  6561 driver19b
SOCKET_NETWORK socket1323  6562 driver19c
SOCKET_NETWORK socket1324  6563 driver19d
SOCKET_NETWORK socket1325  6564 driver19e
SOCKET_NETWORK socket1326  6565 driver19f
SOCKET_NETWORK socket1327  6566 driver19a
SOCKET_NETWORK socket1328  6567 driver19b
SOCKET_NETWORK socket1329  6568 driver19c
SOCKET_NETWORK socket1330  6569 driver19d
SOCKET_NETWORK socket1331  6570 driver20a
SOCKET_NETWORK socket1332  6571 driver20b
SOCKET_NETWORK socket1333  6572 driver20c
SOCKET_NETWORK socket1334  6573 driver20d
SOCKET_NETWORK socket1335  6574 driver20e
SOCKET_NETWORK socket1336  6575 driver20f
SOCKET_NETWORK socket1337  6576 driver20a
SOCKET_NETWORK socket1338  6577 driver20b
SOCKET_NETWORK socket1339  6578 driver20c
SOCKET_NETWORK socket1340  6579 driver20d
SOCKET_NETWORK socket1341  6580 driver20e
SOCKET_NETWORK socket1342  6581 driver20f
SOCKET_NETWORK socket1343  6582 driver20a
SOCKET_NETWORK socket1344  6583 driver20b
SOCKET_NETWORK socket1345  6584 driver20c
SOCKET_NETWORK socket1346  6585 driver20d
SOCKET_NETWORK socket1347  6586 driver20e
SOCKET_NETWORK socket1348  6587 driver20f
```

SOCKET_NETWORK socket1349 6588 driver20a
SOCKET_NETWORK socket1350 6589 driver20b
SOCKET_NETWORK socket1351 6590 driver20c
SOCKET_NETWORK socket1352 6591 driver20d
SOCKET_NETWORK socket1353 6592 driver20e
SOCKET_NETWORK socket1354 6593 driver20f
SOCKET_NETWORK socket1355 6594 driver20a
SOCKET_NETWORK socket1356 6595 driver20b
SOCKET_NETWORK socket1357 6596 driver20c
SOCKET_NETWORK socket1358 6597 driver20d
SOCKET_NETWORK socket1359 6598 driver20e
SOCKET_NETWORK socket1360 6599 driver20f
SOCKET_NETWORK socket1361 6600 driver20a
SOCKET_NETWORK socket1362 6601 driver20b
SOCKET_NETWORK socket1363 6602 driver20c
SOCKET_NETWORK socket1364 6603 driver20d
SOCKET_NETWORK socket1365 6604 driver20e
SOCKET_NETWORK socket1366 6605 driver20f
SOCKET_NETWORK socket1367 6606 driver20a
SOCKET_NETWORK socket1368 6607 driver20b
SOCKET_NETWORK socket1369 6608 driver20c
SOCKET_NETWORK socket1370 6609 driver20d
SOCKET_NETWORK socket1371 6610 driver20e
SOCKET_NETWORK socket1372 6611 driver20f
SOCKET_NETWORK socket1373 6612 driver20a
SOCKET_NETWORK socket1374 6613 driver20b
SOCKET_NETWORK socket1375 6614 driver20c
SOCKET_NETWORK socket1376 6615 driver20d
SOCKET_NETWORK socket1377 6616 driver20e
SOCKET_NETWORK socket1378 6617 driver20f
SOCKET_NETWORK socket1379 6618 driver20a
SOCKET_NETWORK socket1380 6619 driver20b
SOCKET_NETWORK socket1381 6620 driver20c
SOCKET_NETWORK socket1382 6621 driver20d
SOCKET_NETWORK socket1383 6622 driver20e
SOCKET_NETWORK socket1384 6623 driver20f
SOCKET_NETWORK socket1385 6624 driver20a
SOCKET_NETWORK socket1386 6625 driver20b
SOCKET_NETWORK socket1387 6626 driver20c
SOCKET_NETWORK socket1388 6627 driver20d
SOCKET_NETWORK socket1389 6628 driver20e
SOCKET_NETWORK socket1390 6629 driver20f
SOCKET_NETWORK socket1391 6630 driver20a
SOCKET_NETWORK socket1392 6631 driver20b
SOCKET_NETWORK socket1393 6632 driver20c
SOCKET_NETWORK socket1394 6633 driver20d
SOCKET_NETWORK socket1395 6634 driver20e
SOCKET_NETWORK socket1396 6635 driver20f
SOCKET_NETWORK socket1397 6636 driver20a
SOCKET_NETWORK socket1398 6637 driver20b
SOCKET_NETWORK socket1399 6638 driver20c
SOCKET_NETWORK socket1400 6639 driver20d
SOCKET_NETWORK socket1401 6500 driver21a
SOCKET_NETWORK socket1402 6501 driver21b
SOCKET_NETWORK socket1403 6502 driver21c
SOCKET_NETWORK socket1404 6503 driver21d
SOCKET_NETWORK socket1405 6504 driver21e
SOCKET_NETWORK socket1406 6505 driver21f
SOCKET_NETWORK socket1407 6506 driver21a
SOCKET_NETWORK socket1408 6507 driver21b
SOCKET_NETWORK socket1409 6508 driver21c
SOCKET_NETWORK socket1410 6509 driver21d
SOCKET_NETWORK socket1411 6510 driver21e
SOCKET_NETWORK socket1412 6511 driver21f
SOCKET_NETWORK socket1413 6512 driver21a
SOCKET_NETWORK socket1414 6513 driver21b
SOCKET_NETWORK socket1415 6514 driver21c
SOCKET_NETWORK socket1416 6515 driver21d
SOCKET_NETWORK socket1417 6516 driver21e
SOCKET_NETWORK socket1418 6517 driver21f
SOCKET_NETWORK socket1419 6518 driver21a
SOCKET_NETWORK socket1420 6519 driver21b
SOCKET_NETWORK socket1421 6520 driver21c
SOCKET_NETWORK socket1422 6521 driver21d
SOCKET_NETWORK socket1423 6522 driver21e
SOCKET_NETWORK socket1424 6523 driver21f
SOCKET_NETWORK socket1425 6524 driver21a
SOCKET_NETWORK socket1426 6525 driver21b
SOCKET_NETWORK socket1427 6526 driver21c
SOCKET_NETWORK socket1428 6527 driver21d
SOCKET_NETWORK socket1429 6528 driver21e
SOCKET_NETWORK socket1430 6529 driver21f
SOCKET_NETWORK socket1431 6530 driver21a
SOCKET_NETWORK socket1432 6531 driver21b
SOCKET_NETWORK socket1433 6532 driver21c
SOCKET_NETWORK socket1434 6533 driver21d
SOCKET_NETWORK socket1435 6534 driver21e
SOCKET_NETWORK socket1436 6535 driver21f
SOCKET_NETWORK socket1437 6536 driver21a
SOCKET_NETWORK socket1438 6537 driver21b
SOCKET_NETWORK socket1439 6538 driver21c
SOCKET_NETWORK socket1440 6539 driver21d
SOCKET_NETWORK socket1441 6540 driver21e
SOCKET_NETWORK socket1442 6541 driver21f
SOCKET_NETWORK socket1443 6542 driver21a
SOCKET_NETWORK socket1444 6543 driver21b
SOCKET_NETWORK socket1445 6544 driver21c
SOCKET_NETWORK socket1446 6545 driver21d
SOCKET_NETWORK socket1447 6546 driver21e
SOCKET_NETWORK socket1448 6547 driver21f
SOCKET_NETWORK socket1449 6548 driver21a
SOCKET_NETWORK socket1450 6549 driver21b
SOCKET_NETWORK socket1451 6550 driver21c
SOCKET_NETWORK socket1452 6551 driver21d
SOCKET_NETWORK socket1453 6552 driver21e
SOCKET_NETWORK socket1454 6553 driver21f
SOCKET_NETWORK socket1455 6554 driver21a
SOCKET_NETWORK socket1456 6555 driver21b
SOCKET_NETWORK socket1457 6556 driver21c
SOCKET_NETWORK socket1458 6557 driver21d
SOCKET_NETWORK socket1459 6558 driver21e
SOCKET_NETWORK socket1460 6559 driver21f
SOCKET_NETWORK socket1461 6560 driver21a
SOCKET_NETWORK socket1462 6561 driver21b
SOCKET_NETWORK socket1463 6562 driver21c
SOCKET_NETWORK socket1464 6563 driver21d
SOCKET_NETWORK socket1465 6564 driver21e
SOCKET_NETWORK socket1466 6565 driver21f
SOCKET_NETWORK socket1467 6566 driver21a
SOCKET_NETWORK socket1468 6567 driver21b
SOCKET_NETWORK socket1469 6568 driver21c
SOCKET_NETWORK socket1470 6569 driver21d
SOCKET_NETWORK socket1471 6570 driver22a
SOCKET_NETWORK socket1472 6571 driver22b
SOCKET_NETWORK socket1473 6572 driver22c
SOCKET_NETWORK socket1474 6573 driver22d
SOCKET_NETWORK socket1475 6574 driver22e
SOCKET_NETWORK socket1476 6575 driver22f
SOCKET_NETWORK socket1477 6576 driver22a
SOCKET_NETWORK socket1478 6577 driver22b
SOCKET_NETWORK socket1479 6578 driver22c
SOCKET_NETWORK socket1480 6579 driver22d
SOCKET_NETWORK socket1481 6580 driver22e
SOCKET_NETWORK socket1482 6581 driver22f
SOCKET_NETWORK socket1483 6582 driver22a
SOCKET_NETWORK socket1484 6583 driver22b
SOCKET_NETWORK socket1485 6584 driver22c
SOCKET_NETWORK socket1486 6585 driver22d
SOCKET_NETWORK socket1487 6586 driver22e
SOCKET_NETWORK socket1488 6587 driver22f
SOCKET_NETWORK socket1489 6588 driver22a
SOCKET_NETWORK socket1490 6589 driver22b
SOCKET_NETWORK socket1491 6590 driver22c
SOCKET_NETWORK socket1492 6591 driver22d
SOCKET_NETWORK socket1493 6592 driver22e
SOCKET_NETWORK socket1494 6593 driver22f
SOCKET_NETWORK socket1495 6594 driver22a
SOCKET_NETWORK socket1496 6595 driver22b
SOCKET_NETWORK socket1497 6596 driver22c
SOCKET_NETWORK socket1498 6597 driver22d
SOCKET_NETWORK socket1499 6598 driver22e
SOCKET_NETWORK socket1500 6599 driver22f
SOCKET_NETWORK socket1501 6600 driver22a
SOCKET_NETWORK socket1502 6601 driver22b
SOCKET_NETWORK socket1503 6602 driver22c
SOCKET_NETWORK socket1504 6603 driver22d
SOCKET_NETWORK socket1505 6604 driver22e
SOCKET_NETWORK socket1506 6605 driver22f
SOCKET_NETWORK socket1507 6606 driver22a
SOCKET_NETWORK socket1508 6607 driver22b
SOCKET_NETWORK socket1509 6608 driver22c
SOCKET_NETWORK socket1510 6609 driver22d
SOCKET_NETWORK socket1511 6610 driver22e
SOCKET_NETWORK socket1512 6611 driver22f
SOCKET_NETWORK socket1513 6612 driver22a
SOCKET_NETWORK socket1514 6613 driver22b
SOCKET_NETWORK socket1515 6614 driver22c
SOCKET_NETWORK socket1516 6615 driver22d
SOCKET_NETWORK socket1517 6616 driver22e
SOCKET_NETWORK socket1518 6617 driver22f
SOCKET_NETWORK socket1519 6618 driver22a
SOCKET_NETWORK socket1520 6619 driver22b
SOCKET_NETWORK socket1521 6620 driver22c
SOCKET_NETWORK socket1522 6621 driver22d
SOCKET_NETWORK socket1523 6622 driver22e
SOCKET_NETWORK socket1524 6623 driver22f
SOCKET_NETWORK socket1525 6624 driver22a
SOCKET_NETWORK socket1526 6625 driver22b
SOCKET_NETWORK socket1527 6626 driver22c
SOCKET_NETWORK socket1528 6627 driver22d
SOCKET_NETWORK socket1529 6628 driver22e
SOCKET_NETWORK socket1530 6629 driver22f
SOCKET_NETWORK socket1531 6630 driver22a
SOCKET_NETWORK socket1532 6631 driver22b
SOCKET_NETWORK socket1533 6632 driver22c
SOCKET_NETWORK socket1534 6633 driver22d
SOCKET_NETWORK socket1535 6634 driver22e
SOCKET_NETWORK socket1536 6635 driver22f
SOCKET_NETWORK socket1537 6636 driver22a
SOCKET_NETWORK socket1538 6637 driver22b
SOCKET_NETWORK socket1539 6638 driver22c
SOCKET_NETWORK socket1540 6639 driver22d
SOCKET_NETWORK socket1541 6500 driver23a
SOCKET_NETWORK socket1542 6501 driver23b
SOCKET_NETWORK socket1543 6502 driver23c
SOCKET_NETWORK socket1544 6503 driver23d
SOCKET_NETWORK socket1545 6504 driver23e
SOCKET_NETWORK socket1546 6505 driver23f

```
SOCKET_NETWORK socket1547  6506 driver23a
SOCKET_NETWORK socket1548  6507 driver23b
SOCKET_NETWORK socket1549  6508 driver23c
SOCKET_NETWORK socket1550  6509 driver23d
SOCKET_NETWORK socket1551  6510 driver23e
SOCKET_NETWORK socket1552  6511 driver23f
SOCKET_NETWORK socket1553  6512 driver23a
SOCKET_NETWORK socket1554  6513 driver23b
SOCKET_NETWORK socket1555  6514 driver23c
SOCKET_NETWORK socket1556  6515 driver23d
SOCKET_NETWORK socket1557  6516 driver23e
SOCKET_NETWORK socket1558  6517 driver23f
SOCKET_NETWORK socket1559  6518 driver23a
SOCKET_NETWORK socket1560  6519 driver23b
SOCKET_NETWORK socket1561  6520 driver23c
SOCKET_NETWORK socket1562  6521 driver23d
SOCKET_NETWORK socket1563  6522 driver23e
SOCKET_NETWORK socket1564  6523 driver23f
SOCKET_NETWORK socket1565  6524 driver23a
SOCKET_NETWORK socket1566  6525 driver23b
SOCKET_NETWORK socket1567  6526 driver23c
SOCKET_NETWORK socket1568  6527 driver23d
SOCKET_NETWORK socket1569  6528 driver23e
SOCKET_NETWORK socket1570  6529 driver23f
SOCKET_NETWORK socket1571  6530 driver23a
SOCKET_NETWORK socket1572  6531 driver23b
SOCKET_NETWORK socket1573  6532 driver23c
SOCKET_NETWORK socket1574  6533 driver23d
SOCKET_NETWORK socket1575  6534 driver23e
SOCKET_NETWORK socket1576  6535 driver23f
SOCKET_NETWORK socket1577  6536 driver23a
SOCKET_NETWORK socket1578  6537 driver23b
SOCKET_NETWORK socket1579  6538 driver23c
SOCKET_NETWORK socket1580  6539 driver23d
SOCKET_NETWORK socket1581  6540 driver23e
SOCKET_NETWORK socket1582  6541 driver23f
SOCKET_NETWORK socket1583  6542 driver23a
SOCKET_NETWORK socket1584  6543 driver23b
SOCKET_NETWORK socket1585  6544 driver23c
SOCKET_NETWORK socket1586  6545 driver23d
SOCKET_NETWORK socket1587  6546 driver23e
SOCKET_NETWORK socket1588  6547 driver23f
SOCKET_NETWORK socket1589  6548 driver23a
SOCKET_NETWORK socket1590  6549 driver23b
SOCKET_NETWORK socket1591  6550 driver23c
SOCKET_NETWORK socket1592  6551 driver23d
SOCKET_NETWORK socket1593  6552 driver23e
SOCKET_NETWORK socket1594  6553 driver23f
SOCKET_NETWORK socket1595  6554 driver23a
SOCKET_NETWORK socket1596  6555 driver23b
SOCKET_NETWORK socket1597  6556 driver23c
SOCKET_NETWORK socket1598  6557 driver23d
SOCKET_NETWORK socket1599  6558 driver23e
SOCKET_NETWORK socket1600  6559 driver23f
SOCKET_NETWORK socket1601  6560 driver23a
SOCKET_NETWORK socket1602  6561 driver23b
SOCKET_NETWORK socket1603  6562 driver23c
SOCKET_NETWORK socket1604  6563 driver23d
SOCKET_NETWORK socket1605  6564 driver23e
SOCKET_NETWORK socket1606  6565 driver23f
SOCKET_NETWORK socket1607  6566 driver23a
SOCKET_NETWORK socket1608  6567 driver23b
SOCKET_NETWORK socket1609  6568 driver23c
SOCKET_NETWORK socket1610  6569 driver23d
SOCKET_NETWORK socket1611  6570 driver24a
SOCKET_NETWORK socket1612  6571 driver24b
SOCKET_NETWORK socket1613  6572 driver24c
SOCKET_NETWORK socket1614  6573 driver24d
SOCKET_NETWORK socket1615  6574 driver24e
SOCKET_NETWORK socket1616  6575 driver24f
SOCKET_NETWORK socket1617  6576 driver24a
SOCKET_NETWORK socket1618  6577 driver24b
SOCKET_NETWORK socket1619  6578 driver24c
SOCKET_NETWORK socket1620  6579 driver24d
SOCKET_NETWORK socket1621  6580 driver24e
SOCKET_NETWORK socket1622  6581 driver24f
SOCKET_NETWORK socket1623  6582 driver24a
SOCKET_NETWORK socket1624  6583 driver24b
SOCKET_NETWORK socket1625  6584 driver24c
SOCKET_NETWORK socket1626  6585 driver24d
SOCKET_NETWORK socket1627  6586 driver24e
SOCKET_NETWORK socket1628  6587 driver24f
SOCKET_NETWORK socket1629  6588 driver24a
SOCKET_NETWORK socket1630  6589 driver24b
SOCKET_NETWORK socket1631  6590 driver24c
SOCKET_NETWORK socket1632  6591 driver24d
SOCKET_NETWORK socket1633  6592 driver24e
SOCKET_NETWORK socket1634  6593 driver24f
SOCKET_NETWORK socket1635  6594 driver24a
SOCKET_NETWORK socket1636  6595 driver24b
SOCKET_NETWORK socket1637  6596 driver24c
SOCKET_NETWORK socket1638  6597 driver24d
SOCKET_NETWORK socket1639  6598 driver24e
SOCKET_NETWORK socket1640  6599 driver24f
SOCKET_NETWORK socket1641  6600 driver24a
SOCKET_NETWORK socket1642  6601 driver24b
SOCKET_NETWORK socket1643  6602 driver24c
SOCKET_NETWORK socket1644  6603 driver24d
SOCKET_NETWORK socket1645  6604 driver24e
SOCKET_NETWORK socket1646  6605 driver24f
SOCKET_NETWORK socket1647  6606 driver24a
SOCKET_NETWORK socket1648  6607 driver24b
SOCKET_NETWORK socket1649  6608 driver24c
SOCKET_NETWORK socket1650  6609 driver24d
SOCKET_NETWORK socket1651  6610 driver24e
SOCKET_NETWORK socket1652  6611 driver24f
SOCKET_NETWORK socket1653  6612 driver24a
SOCKET_NETWORK socket1654  6613 driver24b
SOCKET_NETWORK socket1655  6614 driver24c
SOCKET_NETWORK socket1656  6615 driver24d
SOCKET_NETWORK socket1657  6616 driver24e
SOCKET_NETWORK socket1658  6617 driver24f
SOCKET_NETWORK socket1659  6618 driver24a
SOCKET_NETWORK socket1660  6619 driver24b
SOCKET_NETWORK socket1661  6620 driver24c
SOCKET_NETWORK socket1662  6621 driver24d
SOCKET_NETWORK socket1663  6622 driver24e
SOCKET_NETWORK socket1664  6623 driver24f
SOCKET_NETWORK socket1665  6624 driver24a
SOCKET_NETWORK socket1666  6625 driver24b
SOCKET_NETWORK socket1667  6626 driver24c
SOCKET_NETWORK socket1668  6627 driver24d
SOCKET_NETWORK socket1669  6628 driver24e
SOCKET_NETWORK socket1670  6629 driver24f
SOCKET_NETWORK socket1671  6630 driver24a
SOCKET_NETWORK socket1672  6631 driver24b
SOCKET_NETWORK socket1673  6632 driver24c
SOCKET_NETWORK socket1674  6633 driver24d
SOCKET_NETWORK socket1675  6634 driver24e
SOCKET_NETWORK socket1676  6635 driver24f
SOCKET_NETWORK socket1677  6636 driver24a
SOCKET_NETWORK socket1678  6637 driver24b
SOCKET_NETWORK socket1679  6638 driver24c
SOCKET_NETWORK socket1680  6639 driver24d
#elif MASTER_NUM5
SOCKET_NETWORK socket1681  6500 driver25a
SOCKET_NETWORK socket1682  6501 driver25b
SOCKET_NETWORK socket1683  6502 driver25c
SOCKET_NETWORK socket1684  6503 driver25d
SOCKET_NETWORK socket1685  6504 driver25e
SOCKET_NETWORK socket1686  6505 driver25f
SOCKET_NETWORK socket1687  6506 driver25a
SOCKET_NETWORK socket1688  6507 driver25b
SOCKET_NETWORK socket1689  6508 driver25c
SOCKET_NETWORK socket1690  6509 driver25d
SOCKET_NETWORK socket1691  6510 driver25e
SOCKET_NETWORK socket1692  6511 driver25f
SOCKET_NETWORK socket1693  6512 driver25a
SOCKET_NETWORK socket1694  6513 driver25b
SOCKET_NETWORK socket1695  6514 driver25c
SOCKET_NETWORK socket1696  6515 driver25d
SOCKET_NETWORK socket1697  6516 driver25e
SOCKET_NETWORK socket1698  6517 driver25f
SOCKET_NETWORK socket1699  6518 driver25a
SOCKET_NETWORK socket1700  6519 driver25b
SOCKET_NETWORK socket1701  6520 driver25c
SOCKET_NETWORK socket1702  6521 driver25d
SOCKET_NETWORK socket1703  6522 driver25e
SOCKET_NETWORK socket1704  6523 driver25f
SOCKET_NETWORK socket1705  6524 driver25a
SOCKET_NETWORK socket1706  6525 driver25b
SOCKET_NETWORK socket1707  6526 driver25c
SOCKET_NETWORK socket1708  6527 driver25d
SOCKET_NETWORK socket1709  6528 driver25e
SOCKET_NETWORK socket1710  6529 driver25f
SOCKET_NETWORK socket1711  6530 driver25a
SOCKET_NETWORK socket1712  6531 driver25b
SOCKET_NETWORK socket1713  6532 driver25c
SOCKET_NETWORK socket1714  6533 driver25d
SOCKET_NETWORK socket1715  6534 driver25e
SOCKET_NETWORK socket1716  6535 driver25f
SOCKET_NETWORK socket1717  6536 driver25a
SOCKET_NETWORK socket1718  6537 driver25b
SOCKET_NETWORK socket1719  6538 driver25c
SOCKET_NETWORK socket1720  6539 driver25d
SOCKET_NETWORK socket1721  6540 driver25e
SOCKET_NETWORK socket1722  6541 driver25f
SOCKET_NETWORK socket1723  6542 driver25a
SOCKET_NETWORK socket1724  6543 driver25b
SOCKET_NETWORK socket1725  6544 driver25c
SOCKET_NETWORK socket1726  6545 driver25d
SOCKET_NETWORK socket1727  6546 driver25e
SOCKET_NETWORK socket1728  6547 driver25f
SOCKET_NETWORK socket1729  6548 driver25a
SOCKET_NETWORK socket1730  6549 driver25b
SOCKET_NETWORK socket1731  6550 driver25c
SOCKET_NETWORK socket1732  6551 driver25d
SOCKET_NETWORK socket1733  6552 driver25e
SOCKET_NETWORK socket1734  6553 driver25f
SOCKET_NETWORK socket1735  6554 driver25a
SOCKET_NETWORK socket1736  6555 driver25b
SOCKET_NETWORK socket1737  6556 driver25c
SOCKET_NETWORK socket1738  6557 driver25d
SOCKET_NETWORK socket1739  6558 driver25e
SOCKET_NETWORK socket1740  6559 driver25f
SOCKET_NETWORK socket1741  6560 driver25a
SOCKET_NETWORK socket1742  6561 driver25b
```

SOCKET_NETWORK	socket1743	6562 driver25c
SOCKET_NETWORK	socket1744	6563 driver25d
SOCKET_NETWORK	socket1745	6564 driver25e
SOCKET_NETWORK	socket1746	6565 driver25f
SOCKET_NETWORK	socket1747	6566 driver25a
SOCKET_NETWORK	socket1748	6567 driver25b
SOCKET_NETWORK	socket1749	6568 driver25c
SOCKET_NETWORK	socket1750	6569 driver25d
SOCKET_NETWORK	socket1751	6570 driver26a
SOCKET_NETWORK	socket1752	6571 driver26b
SOCKET_NETWORK	socket1753	6572 driver26c
SOCKET_NETWORK	socket1754	6573 driver26d
SOCKET_NETWORK	socket1755	6574 driver26e
SOCKET_NETWORK	socket1756	6575 driver26f
SOCKET_NETWORK	socket1757	6576 driver26a
SOCKET_NETWORK	socket1758	6577 driver26b
SOCKET_NETWORK	socket1759	6578 driver26c
SOCKET_NETWORK	socket1760	6579 driver26d
SOCKET_NETWORK	socket1761	6580 driver26e
SOCKET_NETWORK	socket1762	6581 driver26f
SOCKET_NETWORK	socket1763	6582 driver26a
SOCKET_NETWORK	socket1764	6583 driver26b
SOCKET_NETWORK	socket1765	6584 driver26c
SOCKET_NETWORK	socket1766	6585 driver26d
SOCKET_NETWORK	socket1767	6586 driver26e
SOCKET_NETWORK	socket1768	6587 driver26f
SOCKET_NETWORK	socket1769	6588 driver26a
SOCKET_NETWORK	socket1770	6589 driver26b
SOCKET_NETWORK	socket1771	6590 driver26c
SOCKET_NETWORK	socket1772	6591 driver26d
SOCKET_NETWORK	socket1773	6592 driver26e
SOCKET_NETWORK	socket1774	6593 driver26f
SOCKET_NETWORK	socket1775	6594 driver26a
SOCKET_NETWORK	socket1776	6595 driver26b
SOCKET_NETWORK	socket1777	6596 driver26c
SOCKET_NETWORK	socket1778	6597 driver26d
SOCKET_NETWORK	socket1779	6598 driver26e
SOCKET_NETWORK	socket1780	6599 driver26f
SOCKET_NETWORK	socket1781	6600 driver26a
SOCKET_NETWORK	socket1782	6601 driver26b
SOCKET_NETWORK	socket1783	6602 driver26c
SOCKET_NETWORK	socket1784	6603 driver26d
SOCKET_NETWORK	socket1785	6604 driver26e
SOCKET_NETWORK	socket1786	6605 driver26f
SOCKET_NETWORK	socket1787	6606 driver26a
SOCKET_NETWORK	socket1788	6607 driver26b
SOCKET_NETWORK	socket1789	6608 driver26c
SOCKET_NETWORK	socket1790	6609 driver26d
SOCKET_NETWORK	socket1791	6610 driver26e
SOCKET_NETWORK	socket1792	6611 driver26f
SOCKET_NETWORK	socket1793	6612 driver26a
SOCKET_NETWORK	socket1794	6613 driver26b
SOCKET_NETWORK	socket1795	6614 driver26c
SOCKET_NETWORK	socket1796	6615 driver26d
SOCKET_NETWORK	socket1797	6616 driver26e
SOCKET_NETWORK	socket1798	6617 driver26f
SOCKET_NETWORK	socket1799	6618 driver26a
SOCKET_NETWORK	socket1800	6619 driver26b
SOCKET_NETWORK	socket1801	6620 driver26c
SOCKET_NETWORK	socket1802	6621 driver26d
SOCKET_NETWORK	socket1803	6622 driver26e
SOCKET_NETWORK	socket1804	6623 driver26f
SOCKET_NETWORK	socket1805	6624 driver26a
SOCKET_NETWORK	socket1806	6625 driver26b
SOCKET_NETWORK	socket1807	6626 driver26c
SOCKET_NETWORK	socket1808	6627 driver26d
SOCKET_NETWORK	socket1809	6628 driver26e
SOCKET_NETWORK	socket1810	6629 driver26f
SOCKET_NETWORK	socket1811	6630 driver26a
SOCKET_NETWORK	socket1812	6631 driver26b
SOCKET_NETWORK	socket1813	6632 driver26c
SOCKET_NETWORK	socket1814	6633 driver26d
SOCKET_NETWORK	socket1815	6634 driver26e
SOCKET_NETWORK	socket1816	6635 driver26f
SOCKET_NETWORK	socket1817	6636 driver26a
SOCKET_NETWORK	socket1818	6637 driver26b
SOCKET_NETWORK	socket1819	6638 driver26c
SOCKET_NETWORK	socket1820	6639 driver26d
SOCKET_NETWORK	socket1821	6500 driver27a
SOCKET_NETWORK	socket1822	6501 driver27b
SOCKET_NETWORK	socket1823	6502 driver27c
SOCKET_NETWORK	socket1824	6503 driver27d
SOCKET_NETWORK	socket1825	6504 driver27e
SOCKET_NETWORK	socket1826	6505 driver27f
SOCKET_NETWORK	socket1827	6506 driver27a
SOCKET_NETWORK	socket1828	6507 driver27b
SOCKET_NETWORK	socket1829	6508 driver27c
SOCKET_NETWORK	socket1830	6509 driver27d
SOCKET_NETWORK	socket1831	6510 driver27e
SOCKET_NETWORK	socket1832	6511 driver27f
SOCKET_NETWORK	socket1833	6512 driver27a
SOCKET_NETWORK	socket1834	6513 driver27b
SOCKET_NETWORK	socket1835	6514 driver27c
SOCKET_NETWORK	socket1836	6515 driver27d
SOCKET_NETWORK	socket1837	6516 driver27e
SOCKET_NETWORK	socket1838	6517 driver27f
SOCKET_NETWORK	socket1839	6518 driver27a
SOCKET_NETWORK	socket1840	6519 driver27b
SOCKET_NETWORK	socket1841	6520 driver27c
SOCKET_NETWORK	socket1842	6521 driver27d
SOCKET_NETWORK	socket1843	6522 driver27e
SOCKET_NETWORK	socket1844	6523 driver27f
SOCKET_NETWORK	socket1845	6524 driver27a
SOCKET_NETWORK	socket1846	6525 driver27b
SOCKET_NETWORK	socket1847	6526 driver27c
SOCKET_NETWORK	socket1848	6527 driver27d
SOCKET_NETWORK	socket1849	6528 driver27e
SOCKET_NETWORK	socket1850	6529 driver27f
SOCKET_NETWORK	socket1851	6530 driver27a
SOCKET_NETWORK	socket1852	6531 driver27b
SOCKET_NETWORK	socket1853	6532 driver27c
SOCKET_NETWORK	socket1854	6533 driver27d
SOCKET_NETWORK	socket1855	6534 driver27e
SOCKET_NETWORK	socket1856	6535 driver27f
SOCKET_NETWORK	socket1857	6536 driver27a
SOCKET_NETWORK	socket1858	6537 driver27b
SOCKET_NETWORK	socket1859	6538 driver27c
SOCKET_NETWORK	socket1860	6539 driver27d
SOCKET_NETWORK	socket1861	6540 driver27e
SOCKET_NETWORK	socket1862	6541 driver27f
SOCKET_NETWORK	socket1863	6542 driver27a
SOCKET_NETWORK	socket1864	6543 driver27b
SOCKET_NETWORK	socket1865	6544 driver27c
SOCKET_NETWORK	socket1866	6545 driver27d
SOCKET_NETWORK	socket1867	6546 driver27e
SOCKET_NETWORK	socket1868	6547 driver27f
SOCKET_NETWORK	socket1869	6548 driver27a
SOCKET_NETWORK	socket1870	6549 driver27b
SOCKET_NETWORK	socket1871	6550 driver27c
SOCKET_NETWORK	socket1872	6551 driver27d
SOCKET_NETWORK	socket1873	6552 driver27e
SOCKET_NETWORK	socket1874	6553 driver27f
SOCKET_NETWORK	socket1875	6554 driver27a
SOCKET_NETWORK	socket1876	6555 driver27b
SOCKET_NETWORK	socket1877	6556 driver27c
SOCKET_NETWORK	socket1878	6557 driver27d
SOCKET_NETWORK	socket1879	6558 driver27e
SOCKET_NETWORK	socket1880	6559 driver27f
SOCKET_NETWORK	socket1881	6560 driver27a
SOCKET_NETWORK	socket1882	6561 driver27b
SOCKET_NETWORK	socket1883	6562 driver27c
SOCKET_NETWORK	socket1884	6563 driver27d
SOCKET_NETWORK	socket1885	6564 driver27e
SOCKET_NETWORK	socket1886	6565 driver27f
SOCKET_NETWORK	socket1887	6566 driver27a
SOCKET_NETWORK	socket1888	6567 driver27b
SOCKET_NETWORK	socket1889	6568 driver27c
SOCKET_NETWORK	socket1890	6569 driver27d
SOCKET_NETWORK	socket1891	6570 driver28a
SOCKET_NETWORK	socket1892	6571 driver28b
SOCKET_NETWORK	socket1893	6572 driver28c
SOCKET_NETWORK	socket1894	6573 driver28d
SOCKET_NETWORK	socket1895	6574 driver28e
SOCKET_NETWORK	socket1896	6575 driver28f
SOCKET_NETWORK	socket1897	6576 driver28a
SOCKET_NETWORK	socket1898	6577 driver28b
SOCKET_NETWORK	socket1899	6578 driver28c
SOCKET_NETWORK	socket1900	6579 driver28d
SOCKET_NETWORK	socket1901	6580 driver28e
SOCKET_NETWORK	socket1902	6581 driver28f
SOCKET_NETWORK	socket1903	6582 driver28a
SOCKET_NETWORK	socket1904	6583 driver28b
SOCKET_NETWORK	socket1905	6584 driver28c
SOCKET_NETWORK	socket1906	6585 driver28d
SOCKET_NETWORK	socket1907	6586 driver28e
SOCKET_NETWORK	socket1908	6587 driver28f
SOCKET_NETWORK	socket1909	6588 driver28a
SOCKET_NETWORK	socket1910	6589 driver28b
SOCKET_NETWORK	socket1911	6590 driver28c
SOCKET_NETWORK	socket1912	6591 driver28d
SOCKET_NETWORK	socket1913	6592 driver28e
SOCKET_NETWORK	socket1914	6593 driver28f
SOCKET_NETWORK	socket1915	6594 driver28a
SOCKET_NETWORK	socket1916	6595 driver28b
SOCKET_NETWORK	socket1917	6596 driver28c
SOCKET_NETWORK	socket1918	6597 driver28d
SOCKET_NETWORK	socket1919	6598 driver28e
SOCKET_NETWORK	socket1920	6599 driver28f
SOCKET_NETWORK	socket1921	6600 driver28a
SOCKET_NETWORK	socket1922	6601 driver28b
SOCKET_NETWORK	socket1923	6602 driver28c
SOCKET_NETWORK	socket1924	6603 driver28d
SOCKET_NETWORK	socket1925	6604 driver28e
SOCKET_NETWORK	socket1926	6605 driver28f
SOCKET_NETWORK	socket1927	6606 driver28a
SOCKET_NETWORK	socket1928	6607 driver28b
SOCKET_NETWORK	socket1929	6608 driver28c
SOCKET_NETWORK	socket1930	6609 driver28d
SOCKET_NETWORK	socket1931	6610 driver28e
SOCKET_NETWORK	socket1932	6611 driver28f
SOCKET_NETWORK	socket1933	6612 driver28a
SOCKET_NETWORK	socket1934	6613 driver28b
SOCKET_NETWORK	socket1935	6614 driver28c
SOCKET_NETWORK	socket1936	6615 driver28d
SOCKET_NETWORK	socket1937	6616 driver28e
SOCKET_NETWORK	socket1938	6617 driver28f
SOCKET_NETWORK	socket1939	6618 driver28a
SOCKET_NETWORK	socket1940	6619 driver28b

```
SOCKET_NETWORK socket1941  6620 driver28c
SOCKET_NETWORK socket1942  6621 driver28d
SOCKET_NETWORK socket1943  6622 driver28e
SOCKET_NETWORK socket1944  6623 driver28f
SOCKET_NETWORK socket1945  6624 driver28a
SOCKET_NETWORK socket1946  6625 driver28b
SOCKET_NETWORK socket1947  6626 driver28c
SOCKET_NETWORK socket1948  6627 driver28d
SOCKET_NETWORK socket1949  6628 driver28e
SOCKET_NETWORK socket1950  6629 driver28f
SOCKET_NETWORK socket1951  6630 driver28a
SOCKET_NETWORK socket1952  6631 driver28b
SOCKET_NETWORK socket1953  6632 driver28c
SOCKET_NETWORK socket1954  6633 driver28d
SOCKET_NETWORK socket1955  6634 driver28e
SOCKET_NETWORK socket1956  6635 driver28f
SOCKET_NETWORK socket1957  6636 driver28a
SOCKET_NETWORK socket1958  6637 driver28b
SOCKET_NETWORK socket1959  6638 driver28c
SOCKET_NETWORK socket1960  6639 driver28d
SOCKET_NETWORK socket1961  6500 driver29a
SOCKET_NETWORK socket1962  6501 driver29b
SOCKET_NETWORK socket1963  6502 driver29c
SOCKET_NETWORK socket1964  6503 driver29d
SOCKET_NETWORK socket1965  6504 driver29e
SOCKET_NETWORK socket1966  6505 driver29f
SOCKET_NETWORK socket1967  6506 driver29a
SOCKET_NETWORK socket1968  6507 driver29b
SOCKET_NETWORK socket1969  6508 driver29c
SOCKET_NETWORK socket1970  6509 driver29d
SOCKET_NETWORK socket1971  6510 driver29e
SOCKET_NETWORK socket1972  6511 driver29f
SOCKET_NETWORK socket1973  6512 driver29a
SOCKET_NETWORK socket1974  6513 driver29b
SOCKET_NETWORK socket1975  6514 driver29c
SOCKET_NETWORK socket1976  6515 driver29d
SOCKET_NETWORK socket1977  6516 driver29e
SOCKET_NETWORK socket1978  6517 driver29f
SOCKET_NETWORK socket1979  6518 driver29a
SOCKET_NETWORK socket1980  6519 driver29b
SOCKET_NETWORK socket1981  6520 driver29c
SOCKET_NETWORK socket1982  6521 driver29d
SOCKET_NETWORK socket1983  6522 driver29e
SOCKET_NETWORK socket1984  6523 driver29f
SOCKET_NETWORK socket1985  6524 driver29a
SOCKET_NETWORK socket1986  6525 driver29b
SOCKET_NETWORK socket1987  6526 driver29c
SOCKET_NETWORK socket1988  6527 driver29d
SOCKET_NETWORK socket1989  6528 driver29e
SOCKET_NETWORK socket1990  6529 driver29f
SOCKET_NETWORK socket1991  6530 driver29a
SOCKET_NETWORK socket1992  6531 driver29b
SOCKET_NETWORK socket1993  6532 driver29c
SOCKET_NETWORK socket1994  6533 driver29d
SOCKET_NETWORK socket1995  6534 driver29e
SOCKET_NETWORK socket1996  6535 driver29f
SOCKET_NETWORK socket1997  6536 driver29a
SOCKET_NETWORK socket1998  6537 driver29b
SOCKET_NETWORK socket1999  6538 driver29c
SOCKET_NETWORK socket2000  6539 driver29d
SOCKET_NETWORK socket2001  6540 driver29e
SOCKET_NETWORK socket2002  6541 driver29f
SOCKET_NETWORK socket2003  6542 driver29a
SOCKET_NETWORK socket2004  6543 driver29b
SOCKET_NETWORK socket2005  6544 driver29c
SOCKET_NETWORK socket2006  6545 driver29d
SOCKET_NETWORK socket2007  6546 driver29e
SOCKET_NETWORK socket2008  6547 driver29f
SOCKET_NETWORK socket2009  6548 driver29a
SOCKET_NETWORK socket2010  6549 driver29b
SOCKET_NETWORK socket2011  6550 driver29c
SOCKET_NETWORK socket2012  6551 driver29d
SOCKET_NETWORK socket2013  6552 driver29e
SOCKET_NETWORK socket2014  6553 driver29f
SOCKET_NETWORK socket2015  6554 driver29a
SOCKET_NETWORK socket2016  6555 driver29b
SOCKET_NETWORK socket2017  6556 driver29c
SOCKET_NETWORK socket2018  6557 driver29d
SOCKET_NETWORK socket2019  6558 driver29e
SOCKET_NETWORK socket2020  6559 driver29f
SOCKET_NETWORK socket2021  6560 driver29a
SOCKET_NETWORK socket2022  6561 driver29b
SOCKET_NETWORK socket2023  6562 driver29c
SOCKET_NETWORK socket2024  6563 driver29d
SOCKET_NETWORK socket2025  6564 driver29e
SOCKET_NETWORK socket2026  6565 driver29f
SOCKET_NETWORK socket2027  6566 driver29a
SOCKET_NETWORK socket2028  6567 driver29b
SOCKET_NETWORK socket2029  6568 driver29c
SOCKET_NETWORK socket2030  6569 driver29d
SOCKET_NETWORK socket2031  6570 driver30a
SOCKET_NETWORK socket2032  6571 driver30b
SOCKET_NETWORK socket2033  6572 driver30c
SOCKET_NETWORK socket2034  6573 driver30d
SOCKET_NETWORK socket2035  6574 driver30e
SOCKET_NETWORK socket2036  6575 driver30f
SOCKET_NETWORK socket2037  6576 driver30a
SOCKET_NETWORK socket2038  6577 driver30b
SOCKET_NETWORK socket2039  6578 driver30c
SOCKET_NETWORK socket2040  6579 driver30d
SOCKET_NETWORK socket2041  6580 driver30e
SOCKET_NETWORK socket2042  6581 driver30f
SOCKET_NETWORK socket2043  6582 driver30a
SOCKET_NETWORK socket2044  6583 driver30b
SOCKET_NETWORK socket2045  6584 driver30c
SOCKET_NETWORK socket2046  6585 driver30d
SOCKET_NETWORK socket2047  6586 driver30e
SOCKET_NETWORK socket2048  6587 driver30f
SOCKET_NETWORK socket2049  6588 driver30a
SOCKET_NETWORK socket2050  6589 driver30b
SOCKET_NETWORK socket2051  6590 driver30c
SOCKET_NETWORK socket2052  6591 driver30d
SOCKET_NETWORK socket2053  6592 driver30e
SOCKET_NETWORK socket2054  6593 driver30f
SOCKET_NETWORK socket2055  6594 driver30a
SOCKET_NETWORK socket2056  6595 driver30b
SOCKET_NETWORK socket2057  6596 driver30c
SOCKET_NETWORK socket2058  6597 driver30d
SOCKET_NETWORK socket2059  6598 driver30e
SOCKET_NETWORK socket2060  6599 driver30f
SOCKET_NETWORK socket2061  6600 driver30a
SOCKET_NETWORK socket2062  6601 driver30b
SOCKET_NETWORK socket2063  6602 driver30c
SOCKET_NETWORK socket2064  6603 driver30d
SOCKET_NETWORK socket2065  6604 driver30e
SOCKET_NETWORK socket2066  6605 driver30f
SOCKET_NETWORK socket2067  6606 driver30a
SOCKET_NETWORK socket2068  6607 driver30b
SOCKET_NETWORK socket2069  6608 driver30c
SOCKET_NETWORK socket2070  6609 driver30d
SOCKET_NETWORK socket2071  6610 driver30e
SOCKET_NETWORK socket2072  6611 driver30f
SOCKET_NETWORK socket2073  6612 driver30a
SOCKET_NETWORK socket2074  6613 driver30b
SOCKET_NETWORK socket2075  6614 driver30c
SOCKET_NETWORK socket2076  6615 driver30d
SOCKET_NETWORK socket2077  6616 driver30e
SOCKET_NETWORK socket2078  6617 driver30f
SOCKET_NETWORK socket2079  6618 driver30a
SOCKET_NETWORK socket2080  6619 driver30b
SOCKET_NETWORK socket2081  6620 driver30c
SOCKET_NETWORK socket2082  6621 driver30d
SOCKET_NETWORK socket2083  6622 driver30e
SOCKET_NETWORK socket2084  6623 driver30f
SOCKET_NETWORK socket2085  6624 driver30a
SOCKET_NETWORK socket2086  6625 driver30b
SOCKET_NETWORK socket2087  6626 driver30c
SOCKET_NETWORK socket2088  6627 driver30d
SOCKET_NETWORK socket2089  6628 driver30e
SOCKET_NETWORK socket2090  6629 driver30f
SOCKET_NETWORK socket2091  6630 driver30a
SOCKET_NETWORK socket2092  6631 driver30b
SOCKET_NETWORK socket2093  6632 driver30c
SOCKET_NETWORK socket2094  6633 driver30d
SOCKET_NETWORK socket2095  6634 driver30e
SOCKET_NETWORK socket2096  6635 driver30f
SOCKET_NETWORK socket2097  6636 driver30a
SOCKET_NETWORK socket2098  6637 driver30b
SOCKET_NETWORK socket2099  6638 driver30c
SOCKET_NETWORK socket2100  6639 driver30d
#elif MASTER_NUM6
SOCKET_NETWORK socket2101  6500 driver31a
SOCKET_NETWORK socket2102  6501 driver31b
SOCKET_NETWORK socket2103  6502 driver31c
SOCKET_NETWORK socket2104  6503 driver31d
SOCKET_NETWORK socket2105  6504 driver31e
SOCKET_NETWORK socket2106  6505 driver31f
SOCKET_NETWORK socket2107  6506 driver31a
SOCKET_NETWORK socket2108  6507 driver31b
SOCKET_NETWORK socket2109  6508 driver31c
SOCKET_NETWORK socket2110  6509 driver31d
SOCKET_NETWORK socket2111  6510 driver31e
SOCKET_NETWORK socket2112  6511 driver31f
SOCKET_NETWORK socket2113  6512 driver31a
SOCKET_NETWORK socket2114  6513 driver31b
SOCKET_NETWORK socket2115  6514 driver31c
SOCKET_NETWORK socket2116  6515 driver31d
SOCKET_NETWORK socket2117  6516 driver31e
SOCKET_NETWORK socket2118  6517 driver31f
SOCKET_NETWORK socket2119  6518 driver31a
SOCKET_NETWORK socket2120  6519 driver31b
SOCKET_NETWORK socket2121  6520 driver31c
SOCKET_NETWORK socket2122  6521 driver31d
SOCKET_NETWORK socket2123  6522 driver31e
SOCKET_NETWORK socket2124  6523 driver31f
SOCKET_NETWORK socket2125  6524 driver31a
SOCKET_NETWORK socket2126  6525 driver31b
SOCKET_NETWORK socket2127  6526 driver31c
SOCKET_NETWORK socket2128  6527 driver31d
SOCKET_NETWORK socket2129  6528 driver31e
SOCKET_NETWORK socket2130  6529 driver31f
SOCKET_NETWORK socket2131  6530 driver31a
SOCKET_NETWORK socket2132  6531 driver31b
SOCKET_NETWORK socket2133  6532 driver31c
SOCKET_NETWORK socket2134  6533 driver31d
SOCKET_NETWORK socket2135  6534 driver31e
SOCKET_NETWORK socket2136  6535 driver31f
SOCKET_NETWORK socket2137  6536 driver31a
```

SOCKET_NETWORK socket2138 6537 driver31b
SOCKET_NETWORK socket2139 6538 driver31c
SOCKET_NETWORK socket2140 6539 driver31d
SOCKET_NETWORK socket2141 6540 driver31e
SOCKET_NETWORK socket2142 6541 driver31f
SOCKET_NETWORK socket2143 6542 driver31a
SOCKET_NETWORK socket2144 6543 driver31b
SOCKET_NETWORK socket2145 6544 driver31c
SOCKET_NETWORK socket2146 6545 driver31d
SOCKET_NETWORK socket2147 6546 driver31e
SOCKET_NETWORK socket2148 6547 driver31f
SOCKET_NETWORK socket2149 6548 driver31a
SOCKET_NETWORK socket2150 6549 driver31b
SOCKET_NETWORK socket2151 6550 driver31c
SOCKET_NETWORK socket2152 6551 driver31d
SOCKET_NETWORK socket2153 6552 driver31e
SOCKET_NETWORK socket2154 6553 driver31f
SOCKET_NETWORK socket2155 6554 driver31a
SOCKET_NETWORK socket2156 6555 driver31b
SOCKET_NETWORK socket2157 6556 driver31c
SOCKET_NETWORK socket2158 6557 driver31d
SOCKET_NETWORK socket2159 6558 driver31e
SOCKET_NETWORK socket2160 6559 driver31f
SOCKET_NETWORK socket2161 6560 driver31a
SOCKET_NETWORK socket2162 6561 driver31b
SOCKET_NETWORK socket2163 6562 driver31c
SOCKET_NETWORK socket2164 6563 driver31d
SOCKET_NETWORK socket2165 6564 driver31e
SOCKET_NETWORK socket2166 6565 driver31f
SOCKET_NETWORK socket2167 6566 driver31a
SOCKET_NETWORK socket2168 6567 driver31b
SOCKET_NETWORK socket2169 6568 driver31c
SOCKET_NETWORK socket2170 6569 driver31d
SOCKET_NETWORK socket2171 6570 driver32a
SOCKET_NETWORK socket2172 6571 driver32b
SOCKET_NETWORK socket2173 6572 driver32c
SOCKET_NETWORK socket2174 6573 driver32d
SOCKET_NETWORK socket2175 6574 driver32e
SOCKET_NETWORK socket2176 6575 driver32f
SOCKET_NETWORK socket2177 6576 driver32a
SOCKET_NETWORK socket2178 6577 driver32b
SOCKET_NETWORK socket2179 6578 driver32c
SOCKET_NETWORK socket2180 6579 driver32d
SOCKET_NETWORK socket2181 6580 driver32e
SOCKET_NETWORK socket2182 6581 driver32f
SOCKET_NETWORK socket2183 6582 driver32a
SOCKET_NETWORK socket2184 6583 driver32b
SOCKET_NETWORK socket2185 6584 driver32c
SOCKET_NETWORK socket2186 6585 driver32d
SOCKET_NETWORK socket2187 6586 driver32e
SOCKET_NETWORK socket2188 6587 driver32f
SOCKET_NETWORK socket2189 6588 driver32a
SOCKET_NETWORK socket2190 6589 driver32b
SOCKET_NETWORK socket2191 6590 driver32c
SOCKET_NETWORK socket2192 6591 driver32d
SOCKET_NETWORK socket2193 6592 driver32e
SOCKET_NETWORK socket2194 6593 driver32f
SOCKET_NETWORK socket2195 6594 driver32a
SOCKET_NETWORK socket2196 6595 driver32b
SOCKET_NETWORK socket2197 6596 driver32c
SOCKET_NETWORK socket2198 6597 driver32d
SOCKET_NETWORK socket2199 6598 driver32e
SOCKET_NETWORK socket2200 6599 driver32f
SOCKET_NETWORK socket2201 6600 driver32a
SOCKET_NETWORK socket2202 6601 driver32b
SOCKET_NETWORK socket2203 6602 driver32c
SOCKET_NETWORK socket2204 6603 driver32d
SOCKET_NETWORK socket2205 6604 driver32e
SOCKET_NETWORK socket2206 6605 driver32f
SOCKET_NETWORK socket2207 6606 driver32a
SOCKET_NETWORK socket2208 6607 driver32b
SOCKET_NETWORK socket2209 6608 driver32c
SOCKET_NETWORK socket2210 6609 driver32d
SOCKET_NETWORK socket2211 6610 driver32e
SOCKET_NETWORK socket2212 6611 driver32f
SOCKET_NETWORK socket2213 6612 driver32a
SOCKET_NETWORK socket2214 6613 driver32b
SOCKET_NETWORK socket2215 6614 driver32c
SOCKET_NETWORK socket2216 6615 driver32d
SOCKET_NETWORK socket2217 6616 driver32e
SOCKET_NETWORK socket2218 6617 driver32f
SOCKET_NETWORK socket2219 6618 driver32a
SOCKET_NETWORK socket2220 6619 driver32b
SOCKET_NETWORK socket2221 6620 driver32c
SOCKET_NETWORK socket2222 6621 driver32d
SOCKET_NETWORK socket2223 6622 driver32e
SOCKET_NETWORK socket2224 6623 driver32f
SOCKET_NETWORK socket2225 6624 driver32a
SOCKET_NETWORK socket2226 6625 driver32b
SOCKET_NETWORK socket2227 6626 driver32c
SOCKET_NETWORK socket2228 6627 driver32d
SOCKET_NETWORK socket2229 6628 driver32e
SOCKET_NETWORK socket2230 6629 driver32f
SOCKET_NETWORK socket2231 6630 driver32a
SOCKET_NETWORK socket2232 6631 driver32b
SOCKET_NETWORK socket2233 6632 driver32c
SOCKET_NETWORK socket2234 6633 driver32d
SOCKET_NETWORK socket2235 6634 driver32e
SOCKET_NETWORK socket2236 6635 driver32f
SOCKET_NETWORK socket2237 6636 driver32a
SOCKET_NETWORK socket2238 6637 driver32b
SOCKET_NETWORK socket2239 6638 driver32c
SOCKET_NETWORK socket2240 6639 driver32d
SOCKET_NETWORK socket2241 6500 driver33a
SOCKET_NETWORK socket2242 6501 driver33b
SOCKET_NETWORK socket2243 6502 driver33c
SOCKET_NETWORK socket2244 6503 driver33d
SOCKET_NETWORK socket2245 6504 driver33e
SOCKET_NETWORK socket2246 6505 driver33f
SOCKET_NETWORK socket2247 6506 driver33a
SOCKET_NETWORK socket2248 6507 driver33b
SOCKET_NETWORK socket2249 6508 driver33c
SOCKET_NETWORK socket2250 6509 driver33d
SOCKET_NETWORK socket2251 6510 driver33e
SOCKET_NETWORK socket2252 6511 driver33f
SOCKET_NETWORK socket2253 6512 driver33a
SOCKET_NETWORK socket2254 6513 driver33b
SOCKET_NETWORK socket2255 6514 driver33c
SOCKET_NETWORK socket2256 6515 driver33d
SOCKET_NETWORK socket2257 6516 driver33e
SOCKET_NETWORK socket2258 6517 driver33f
SOCKET_NETWORK socket2259 6518 driver33a
SOCKET_NETWORK socket2260 6519 driver33b
SOCKET_NETWORK socket2261 6520 driver33c
SOCKET_NETWORK socket2262 6521 driver33d
SOCKET_NETWORK socket2263 6522 driver33e
SOCKET_NETWORK socket2264 6523 driver33f
SOCKET_NETWORK socket2265 6524 driver33a
SOCKET_NETWORK socket2266 6525 driver33b
SOCKET_NETWORK socket2267 6526 driver33c
SOCKET_NETWORK socket2268 6527 driver33d
SOCKET_NETWORK socket2269 6528 driver33e
SOCKET_NETWORK socket2270 6529 driver33f
SOCKET_NETWORK socket2271 6530 driver33a
SOCKET_NETWORK socket2272 6531 driver33b
SOCKET_NETWORK socket2273 6532 driver33c
SOCKET_NETWORK socket2274 6533 driver33d
SOCKET_NETWORK socket2275 6534 driver33e
SOCKET_NETWORK socket2276 6535 driver33f
SOCKET_NETWORK socket2277 6536 driver33a
SOCKET_NETWORK socket2278 6537 driver33b
SOCKET_NETWORK socket2279 6538 driver33c
SOCKET_NETWORK socket2280 6539 driver33d
SOCKET_NETWORK socket2281 6540 driver33e
SOCKET_NETWORK socket2282 6541 driver33f
SOCKET_NETWORK socket2283 6542 driver33a
SOCKET_NETWORK socket2284 6543 driver33b
SOCKET_NETWORK socket2285 6544 driver33c
SOCKET_NETWORK socket2286 6545 driver33d
SOCKET_NETWORK socket2287 6546 driver33e
SOCKET_NETWORK socket2288 6547 driver33f
SOCKET_NETWORK socket2289 6548 driver33a
SOCKET_NETWORK socket2290 6549 driver33b
SOCKET_NETWORK socket2291 6550 driver33c
SOCKET_NETWORK socket2292 6551 driver33d
SOCKET_NETWORK socket2293 6552 driver33e
SOCKET_NETWORK socket2294 6553 driver33f
SOCKET_NETWORK socket2295 6554 driver33a
SOCKET_NETWORK socket2296 6555 driver33b
SOCKET_NETWORK socket2297 6556 driver33c
SOCKET_NETWORK socket2298 6557 driver33d
SOCKET_NETWORK socket2299 6558 driver33e
SOCKET_NETWORK socket2300 6559 driver33f
SOCKET_NETWORK socket2301 6560 driver33a
SOCKET_NETWORK socket2302 6561 driver33b
SOCKET_NETWORK socket2303 6562 driver33c
SOCKET_NETWORK socket2304 6563 driver33d
SOCKET_NETWORK socket2305 6564 driver33e
SOCKET_NETWORK socket2306 6565 driver33f
SOCKET_NETWORK socket2307 6566 driver33a
SOCKET_NETWORK socket2308 6567 driver33b
SOCKET_NETWORK socket2309 6568 driver33c
SOCKET_NETWORK socket2310 6569 driver33d
SOCKET_NETWORK socket2311 6570 driver34a
SOCKET_NETWORK socket2312 6571 driver34b
SOCKET_NETWORK socket2313 6572 driver34c
SOCKET_NETWORK socket2314 6573 driver34d
SOCKET_NETWORK socket2315 6574 driver34e
SOCKET_NETWORK socket2316 6575 driver34f
SOCKET_NETWORK socket2317 6576 driver34a
SOCKET_NETWORK socket2318 6577 driver34b
SOCKET_NETWORK socket2319 6578 driver34c
SOCKET_NETWORK socket2320 6579 driver34d
SOCKET_NETWORK socket2321 6580 driver34e
SOCKET_NETWORK socket2322 6581 driver34f
SOCKET_NETWORK socket2323 6582 driver34a
SOCKET_NETWORK socket2324 6583 driver34b
SOCKET_NETWORK socket2325 6584 driver34c
SOCKET_NETWORK socket2326 6585 driver34d
SOCKET_NETWORK socket2327 6586 driver34e
SOCKET_NETWORK socket2328 6587 driver34f
SOCKET_NETWORK socket2329 6588 driver34a
SOCKET_NETWORK socket2330 6589 driver34b
SOCKET_NETWORK socket2331 6590 driver34c
SOCKET_NETWORK socket2332 6591 driver34d
SOCKET_NETWORK socket2333 6592 driver34e
SOCKET_NETWORK socket2334 6593 driver34f
SOCKET_NETWORK socket2335 6594 driver34a

```
SOCKET_NETWORK socket2336  6595 driver34b
SOCKET_NETWORK socket2337  6596 driver34c
SOCKET_NETWORK socket2338  6597 driver34d
SOCKET_NETWORK socket2339  6598 driver34e
SOCKET_NETWORK socket2340  6599 driver34f
SOCKET_NETWORK socket2341  6600 driver34a
SOCKET_NETWORK socket2342  6601 driver34b
SOCKET_NETWORK socket2343  6602 driver34c
SOCKET_NETWORK socket2344  6603 driver34d
SOCKET_NETWORK socket2345  6604 driver34e
SOCKET_NETWORK socket2346  6605 driver34f
SOCKET_NETWORK socket2347  6606 driver34a
SOCKET_NETWORK socket2348  6607 driver34b
SOCKET_NETWORK socket2349  6608 driver34c
SOCKET_NETWORK socket2350  6609 driver34d
SOCKET_NETWORK socket2351  6610 driver34e
SOCKET_NETWORK socket2352  6611 driver34f
SOCKET_NETWORK socket2353  6612 driver34a
SOCKET_NETWORK socket2354  6613 driver34b
SOCKET_NETWORK socket2355  6614 driver34c
SOCKET_NETWORK socket2356  6615 driver34d
SOCKET_NETWORK socket2357  6616 driver34e
SOCKET_NETWORK socket2358  6617 driver34f
SOCKET_NETWORK socket2359  6618 driver34a
SOCKET_NETWORK socket2360  6619 driver34b
SOCKET_NETWORK socket2361  6620 driver34c
SOCKET_NETWORK socket2362  6621 driver34d
SOCKET_NETWORK socket2363  6622 driver34e
SOCKET_NETWORK socket2364  6623 driver34f
SOCKET_NETWORK socket2365  6624 driver34a
SOCKET_NETWORK socket2366  6625 driver34b
SOCKET_NETWORK socket2367  6626 driver34c
SOCKET_NETWORK socket2368  6627 driver34d
SOCKET_NETWORK socket2369  6628 driver34e
SOCKET_NETWORK socket2370  6629 driver34f
SOCKET_NETWORK socket2371  6630 driver34a
SOCKET_NETWORK socket2372  6631 driver34b
SOCKET_NETWORK socket2373  6632 driver34c
SOCKET_NETWORK socket2374  6633 driver34d
SOCKET_NETWORK socket2375  6634 driver34e
SOCKET_NETWORK socket2376  6635 driver34f
SOCKET_NETWORK socket2377  6636 driver34a
SOCKET_NETWORK socket2378  6637 driver34b
SOCKET_NETWORK socket2379  6638 driver34c
SOCKET_NETWORK socket2380  6639 driver34d
SOCKET_NETWORK socket2381  6500 driver35a
SOCKET_NETWORK socket2382  6501 driver35b
SOCKET_NETWORK socket2383  6502 driver35c
SOCKET_NETWORK socket2384  6503 driver35d
SOCKET_NETWORK socket2385  6504 driver35e
SOCKET_NETWORK socket2386  6505 driver35f
SOCKET_NETWORK socket2387  6506 driver35a
SOCKET_NETWORK socket2388  6507 driver35b
SOCKET_NETWORK socket2389  6508 driver35c
SOCKET_NETWORK socket2390  6509 driver35d
SOCKET_NETWORK socket2391  6510 driver35e
SOCKET_NETWORK socket2392  6511 driver35f
SOCKET_NETWORK socket2393  6512 driver35a
SOCKET_NETWORK socket2394  6513 driver35b
SOCKET_NETWORK socket2395  6514 driver35c
SOCKET_NETWORK socket2396  6515 driver35d
SOCKET_NETWORK socket2397  6516 driver35e
SOCKET_NETWORK socket2398  6517 driver35f
SOCKET_NETWORK socket2399  6518 driver35a
SOCKET_NETWORK socket2400  6519 driver35b
SOCKET_NETWORK socket2401  6520 driver35c
SOCKET_NETWORK socket2402  6521 driver35d
SOCKET_NETWORK socket2403  6522 driver35e
SOCKET_NETWORK socket2404  6523 driver35f
SOCKET_NETWORK socket2405  6524 driver35a
SOCKET_NETWORK socket2406  6525 driver35b
SOCKET_NETWORK socket2407  6526 driver35c
SOCKET_NETWORK socket2408  6527 driver35d
SOCKET_NETWORK socket2409  6528 driver35e
SOCKET_NETWORK socket2410  6529 driver35f
SOCKET_NETWORK socket2411  6530 driver35a
SOCKET_NETWORK socket2412  6531 driver35b
SOCKET_NETWORK socket2413  6532 driver35c
SOCKET_NETWORK socket2414  6533 driver35d
SOCKET_NETWORK socket2415  6534 driver35e
SOCKET_NETWORK socket2416  6535 driver35f
SOCKET_NETWORK socket2417  6536 driver35a
SOCKET_NETWORK socket2418  6537 driver35b
SOCKET_NETWORK socket2419  6538 driver35c
SOCKET_NETWORK socket2420  6539 driver35d
SOCKET_NETWORK socket2421  6540 driver35e
SOCKET_NETWORK socket2422  6541 driver35f
SOCKET_NETWORK socket2423  6542 driver35a
SOCKET_NETWORK socket2424  6543 driver35b
SOCKET_NETWORK socket2425  6544 driver35c
SOCKET_NETWORK socket2426  6545 driver35d
SOCKET_NETWORK socket2427  6546 driver35e
SOCKET_NETWORK socket2428  6547 driver35f
SOCKET_NETWORK socket2429  6548 driver35a
SOCKET_NETWORK socket2430  6549 driver35b
SOCKET_NETWORK socket2431  6550 driver35c
SOCKET_NETWORK socket2432  6551 driver35d
SOCKET_NETWORK socket2433  6552 driver35e
SOCKET_NETWORK socket2434  6553 driver35f
SOCKET_NETWORK socket2435  6554 driver35a
SOCKET_NETWORK socket2436  6555 driver35b
SOCKET_NETWORK socket2437  6556 driver35c
SOCKET_NETWORK socket2438  6557 driver35d
SOCKET_NETWORK socket2439  6558 driver35e
SOCKET_NETWORK socket2440  6559 driver35f
SOCKET_NETWORK socket2441  6560 driver35a
SOCKET_NETWORK socket2442  6561 driver35b
SOCKET_NETWORK socket2443  6562 driver35c
SOCKET_NETWORK socket2444  6563 driver35d
SOCKET_NETWORK socket2445  6564 driver35e
SOCKET_NETWORK socket2446  6565 driver35f
SOCKET_NETWORK socket2447  6566 driver35a
SOCKET_NETWORK socket2448  6567 driver35b
SOCKET_NETWORK socket2449  6568 driver35c
SOCKET_NETWORK socket2450  6569 driver35d
SOCKET_NETWORK socket2451  6570 driver36a
SOCKET_NETWORK socket2452  6571 driver36b
SOCKET_NETWORK socket2453  6572 driver36c
SOCKET_NETWORK socket2454  6573 driver36d
SOCKET_NETWORK socket2455  6574 driver36e
SOCKET_NETWORK socket2456  6575 driver36f
SOCKET_NETWORK socket2457  6576 driver36a
SOCKET_NETWORK socket2458  6577 driver36b
SOCKET_NETWORK socket2459  6578 driver36c
SOCKET_NETWORK socket2460  6579 driver36d
SOCKET_NETWORK socket2461  6580 driver36e
SOCKET_NETWORK socket2462  6581 driver36f
SOCKET_NETWORK socket2463  6582 driver36a
SOCKET_NETWORK socket2464  6583 driver36b
SOCKET_NETWORK socket2465  6584 driver36c
SOCKET_NETWORK socket2466  6585 driver36d
SOCKET_NETWORK socket2467  6586 driver36e
SOCKET_NETWORK socket2468  6587 driver36f
SOCKET_NETWORK socket2469  6588 driver36a
SOCKET_NETWORK socket2470  6589 driver36b
SOCKET_NETWORK socket2471  6590 driver36c
SOCKET_NETWORK socket2472  6591 driver36d
SOCKET_NETWORK socket2473  6592 driver36e
SOCKET_NETWORK socket2474  6593 driver36f
SOCKET_NETWORK socket2475  6594 driver36a
SOCKET_NETWORK socket2476  6595 driver36b
SOCKET_NETWORK socket2477  6596 driver36c
SOCKET_NETWORK socket2478  6597 driver36d
SOCKET_NETWORK socket2479  6598 driver36e
SOCKET_NETWORK socket2480  6599 driver36f
SOCKET_NETWORK socket2481  6600 driver36a
SOCKET_NETWORK socket2482  6601 driver36b
SOCKET_NETWORK socket2483  6602 driver36c
SOCKET_NETWORK socket2484  6603 driver36d
SOCKET_NETWORK socket2485  6604 driver36e
SOCKET_NETWORK socket2486  6605 driver36f
SOCKET_NETWORK socket2487  6606 driver36a
SOCKET_NETWORK socket2488  6607 driver36b
SOCKET_NETWORK socket2489  6608 driver36c
SOCKET_NETWORK socket2490  6609 driver36d
SOCKET_NETWORK socket2491  6610 driver36e
SOCKET_NETWORK socket2492  6611 driver36f
SOCKET_NETWORK socket2493  6612 driver36a
SOCKET_NETWORK socket2494  6613 driver36b
SOCKET_NETWORK socket2495  6614 driver36c
SOCKET_NETWORK socket2496  6615 driver36d
SOCKET_NETWORK socket2497  6616 driver36e
SOCKET_NETWORK socket2498  6617 driver36f
SOCKET_NETWORK socket2499  6618 driver36a
SOCKET_NETWORK socket2500  6619 driver36b
SOCKET_NETWORK socket2501  6620 driver36c
SOCKET_NETWORK socket2502  6621 driver36d
SOCKET_NETWORK socket2503  6622 driver36e
SOCKET_NETWORK socket2504  6623 driver36f
SOCKET_NETWORK socket2505  6624 driver36a
SOCKET_NETWORK socket2506  6625 driver36b
SOCKET_NETWORK socket2507  6626 driver36c
SOCKET_NETWORK socket2508  6627 driver36d
SOCKET_NETWORK socket2509  6628 driver36e
SOCKET_NETWORK socket2510  6629 driver36f
SOCKET_NETWORK socket2511  6630 driver36a
SOCKET_NETWORK socket2512  6631 driver36b
SOCKET_NETWORK socket2513  6632 driver36c
SOCKET_NETWORK socket2514  6633 driver36d
SOCKET_NETWORK socket2515  6634 driver36e
SOCKET_NETWORK socket2516  6635 driver36f
SOCKET_NETWORK socket2517  6636 driver36a
SOCKET_NETWORK socket2518  6637 driver36b
SOCKET_NETWORK socket2519  6638 driver36c
SOCKET_NETWORK socket2520  6639 driver36d
#elif MASTER_NUM7
SOCKET_NETWORK socket2521  6500 driver37a
SOCKET_NETWORK socket2522  6501 driver37b
SOCKET_NETWORK socket2523  6502 driver37c
SOCKET_NETWORK socket2524  6503 driver37d
SOCKET_NETWORK socket2525  6504 driver37e
SOCKET_NETWORK socket2526  6505 driver37f
SOCKET_NETWORK socket2527  6506 driver37a
SOCKET_NETWORK socket2528  6507 driver37b
SOCKET_NETWORK socket2529  6508 driver37c
SOCKET_NETWORK socket2530  6509 driver37d
SOCKET_NETWORK socket2531  6510 driver37e
SOCKET_NETWORK socket2532  6511 driver37f
```

SOCKET_NETWORK socket2533 6512 driver37a
SOCKET_NETWORK socket2534 6513 driver37b
SOCKET_NETWORK socket2535 6514 driver37c
SOCKET_NETWORK socket2536 6515 driver37d
SOCKET_NETWORK socket2537 6516 driver37e
SOCKET_NETWORK socket2538 6517 driver37f
SOCKET_NETWORK socket2539 6518 driver37a
SOCKET_NETWORK socket2540 6519 driver37b
SOCKET_NETWORK socket2541 6520 driver37c
SOCKET_NETWORK socket2542 6521 driver37d
SOCKET_NETWORK socket2543 6522 driver37e
SOCKET_NETWORK socket2544 6523 driver37f
SOCKET_NETWORK socket2545 6524 driver37a
SOCKET_NETWORK socket2546 6525 driver37b
SOCKET_NETWORK socket2547 6526 driver37c
SOCKET_NETWORK socket2548 6527 driver37d
SOCKET_NETWORK socket2549 6528 driver37e
SOCKET_NETWORK socket2550 6529 driver37f
SOCKET_NETWORK socket2551 6530 driver37a
SOCKET_NETWORK socket2552 6531 driver37b
SOCKET_NETWORK socket2553 6532 driver37c
SOCKET_NETWORK socket2554 6533 driver37d
SOCKET_NETWORK socket2555 6534 driver37e
SOCKET_NETWORK socket2556 6535 driver37f
SOCKET_NETWORK socket2557 6536 driver37a
SOCKET_NETWORK socket2558 6537 driver37b
SOCKET_NETWORK socket2559 6538 driver37c
SOCKET_NETWORK socket2560 6539 driver37d
SOCKET_NETWORK socket2561 6540 driver37e
SOCKET_NETWORK socket2562 6541 driver37f
SOCKET_NETWORK socket2563 6542 driver37a
SOCKET_NETWORK socket2564 6543 driver37b
SOCKET_NETWORK socket2565 6544 driver37c
SOCKET_NETWORK socket2566 6545 driver37d
SOCKET_NETWORK socket2567 6546 driver37e
SOCKET_NETWORK socket2568 6547 driver37f
SOCKET_NETWORK socket2569 6548 driver37a
SOCKET_NETWORK socket2570 6549 driver37b
SOCKET_NETWORK socket2571 6550 driver37c
SOCKET_NETWORK socket2572 6551 driver37d
SOCKET_NETWORK socket2573 6552 driver37e
SOCKET_NETWORK socket2574 6553 driver37f
SOCKET_NETWORK socket2575 6554 driver37a
SOCKET_NETWORK socket2576 6555 driver37b
SOCKET_NETWORK socket2577 6556 driver37c
SOCKET_NETWORK socket2578 6557 driver37d
SOCKET_NETWORK socket2579 6558 driver37e
SOCKET_NETWORK socket2580 6559 driver37f
SOCKET_NETWORK socket2581 6560 driver37a
SOCKET_NETWORK socket2582 6561 driver37b
SOCKET_NETWORK socket2583 6562 driver37c
SOCKET_NETWORK socket2584 6563 driver37d
SOCKET_NETWORK socket2585 6564 driver37e
SOCKET_NETWORK socket2586 6565 driver37f
SOCKET_NETWORK socket2587 6566 driver37a
SOCKET_NETWORK socket2588 6567 driver37b
SOCKET_NETWORK socket2589 6568 driver37c
SOCKET_NETWORK socket2590 6569 driver37d
SOCKET_NETWORK socket2591 6570 driver38a
SOCKET_NETWORK socket2592 6571 driver38b
SOCKET_NETWORK socket2593 6572 driver38c
SOCKET_NETWORK socket2594 6573 driver38d
SOCKET_NETWORK socket2595 6574 driver38e
SOCKET_NETWORK socket2596 6575 driver38f
SOCKET_NETWORK socket2597 6576 driver38a
SOCKET_NETWORK socket2598 6577 driver38b
SOCKET_NETWORK socket2599 6578 driver38c
SOCKET_NETWORK socket2600 6579 driver38d
SOCKET_NETWORK socket2601 6580 driver38e
SOCKET_NETWORK socket2602 6581 driver38f
SOCKET_NETWORK socket2603 6582 driver38a
SOCKET_NETWORK socket2604 6583 driver38b
SOCKET_NETWORK socket2605 6584 driver38c
SOCKET_NETWORK socket2606 6585 driver38d
SOCKET_NETWORK socket2607 6586 driver38e
SOCKET_NETWORK socket2608 6587 driver38f
SOCKET_NETWORK socket2609 6588 driver38a
SOCKET_NETWORK socket2610 6589 driver38b
SOCKET_NETWORK socket2611 6590 driver38c
SOCKET_NETWORK socket2612 6591 driver38d
SOCKET_NETWORK socket2613 6592 driver38e
SOCKET_NETWORK socket2614 6593 driver38f
SOCKET_NETWORK socket2615 6594 driver38a
SOCKET_NETWORK socket2616 6595 driver38b
SOCKET_NETWORK socket2617 6596 driver38c
SOCKET_NETWORK socket2618 6597 driver38d
SOCKET_NETWORK socket2619 6598 driver38e
SOCKET_NETWORK socket2620 6599 driver38f
SOCKET_NETWORK socket2621 6600 driver38a
SOCKET_NETWORK socket2622 6601 driver38b
SOCKET_NETWORK socket2623 6602 driver38c
SOCKET_NETWORK socket2624 6603 driver38d
SOCKET_NETWORK socket2625 6604 driver38e
SOCKET_NETWORK socket2626 6605 driver38f
SOCKET_NETWORK socket2627 6606 driver38a
SOCKET_NETWORK socket2628 6607 driver38b
SOCKET_NETWORK socket2629 6608 driver38c
SOCKET_NETWORK socket2630 6609 driver38d
SOCKET_NETWORK socket2631 6610 driver38e
SOCKET_NETWORK socket2632 6611 driver38f
SOCKET_NETWORK socket2633 6612 driver38a
SOCKET_NETWORK socket2634 6613 driver38b
SOCKET_NETWORK socket2635 6614 driver38c
SOCKET_NETWORK socket2636 6615 driver38d
SOCKET_NETWORK socket2637 6616 driver38e
SOCKET_NETWORK socket2638 6617 driver38f
SOCKET_NETWORK socket2639 6618 driver38a
SOCKET_NETWORK socket2640 6619 driver38b
SOCKET_NETWORK socket2641 6620 driver38c
SOCKET_NETWORK socket2642 6621 driver38d
SOCKET_NETWORK socket2643 6622 driver38e
SOCKET_NETWORK socket2644 6623 driver38f
SOCKET_NETWORK socket2645 6624 driver38a
SOCKET_NETWORK socket2646 6625 driver38b
SOCKET_NETWORK socket2647 6626 driver38c
SOCKET_NETWORK socket2648 6627 driver38d
SOCKET_NETWORK socket2649 6628 driver38e
SOCKET_NETWORK socket2650 6629 driver38f
SOCKET_NETWORK socket2651 6630 driver38a
SOCKET_NETWORK socket2652 6631 driver38b
SOCKET_NETWORK socket2653 6632 driver38c
SOCKET_NETWORK socket2654 6633 driver38d
SOCKET_NETWORK socket2655 6634 driver38e
SOCKET_NETWORK socket2656 6635 driver38f
SOCKET_NETWORK socket2657 6636 driver38a
SOCKET_NETWORK socket2658 6637 driver38b
SOCKET_NETWORK socket2659 6638 driver38c
SOCKET_NETWORK socket2660 6639 driver38d
SOCKET_NETWORK socket2661 6500 driver39a
SOCKET_NETWORK socket2662 6501 driver39b
SOCKET_NETWORK socket2663 6502 driver39c
SOCKET_NETWORK socket2664 6503 driver39d
SOCKET_NETWORK socket2665 6504 driver39e
SOCKET_NETWORK socket2666 6505 driver39f
SOCKET_NETWORK socket2667 6506 driver39a
SOCKET_NETWORK socket2668 6507 driver39b
SOCKET_NETWORK socket2669 6508 driver39c
SOCKET_NETWORK socket2670 6509 driver39d
SOCKET_NETWORK socket2671 6510 driver39e
SOCKET_NETWORK socket2672 6511 driver39f
SOCKET_NETWORK socket2673 6512 driver39a
SOCKET_NETWORK socket2674 6513 driver39b
SOCKET_NETWORK socket2675 6514 driver39c
SOCKET_NETWORK socket2676 6515 driver39d
SOCKET_NETWORK socket2677 6516 driver39e
SOCKET_NETWORK socket2678 6517 driver39f
SOCKET_NETWORK socket2679 6518 driver39a
SOCKET_NETWORK socket2680 6519 driver39b
SOCKET_NETWORK socket2681 6520 driver39c
SOCKET_NETWORK socket2682 6521 driver39d
SOCKET_NETWORK socket2683 6522 driver39e
SOCKET_NETWORK socket2684 6523 driver39f
SOCKET_NETWORK socket2685 6524 driver39a
SOCKET_NETWORK socket2686 6525 driver39b
SOCKET_NETWORK socket2687 6526 driver39c
SOCKET_NETWORK socket2688 6527 driver39d
SOCKET_NETWORK socket2689 6528 driver39e
SOCKET_NETWORK socket2690 6529 driver39f
SOCKET_NETWORK socket2691 6530 driver39a
SOCKET_NETWORK socket2692 6531 driver39b
SOCKET_NETWORK socket2693 6532 driver39c
SOCKET_NETWORK socket2694 6533 driver39d
SOCKET_NETWORK socket2695 6534 driver39e
SOCKET_NETWORK socket2696 6535 driver39f
SOCKET_NETWORK socket2697 6536 driver39a
SOCKET_NETWORK socket2698 6537 driver39b
SOCKET_NETWORK socket2699 6538 driver39c
SOCKET_NETWORK socket2700 6539 driver39d
SOCKET_NETWORK socket2701 6540 driver39e
SOCKET_NETWORK socket2702 6541 driver39f
SOCKET_NETWORK socket2703 6542 driver39a
SOCKET_NETWORK socket2704 6543 driver39b
SOCKET_NETWORK socket2705 6544 driver39c
SOCKET_NETWORK socket2706 6545 driver39d
SOCKET_NETWORK socket2707 6546 driver39e
SOCKET_NETWORK socket2708 6547 driver39f
SOCKET_NETWORK socket2709 6548 driver39a
SOCKET_NETWORK socket2710 6549 driver39b
SOCKET_NETWORK socket2711 6550 driver39c
SOCKET_NETWORK socket2712 6551 driver39d
SOCKET_NETWORK socket2713 6552 driver39e
SOCKET_NETWORK socket2714 6553 driver39f
SOCKET_NETWORK socket2715 6554 driver39a
SOCKET_NETWORK socket2716 6555 driver39b
SOCKET_NETWORK socket2717 6556 driver39c
SOCKET_NETWORK socket2718 6557 driver39d
SOCKET_NETWORK socket2719 6558 driver39e
SOCKET_NETWORK socket2720 6559 driver39f
SOCKET_NETWORK socket2721 6560 driver39a
SOCKET_NETWORK socket2722 6561 driver39b
SOCKET_NETWORK socket2723 6562 driver39c
SOCKET_NETWORK socket2724 6563 driver39d
SOCKET_NETWORK socket2725 6564 driver39e
SOCKET_NETWORK socket2726 6565 driver39f
SOCKET_NETWORK socket2727 6566 driver39a
SOCKET_NETWORK socket2728 6567 driver39b
SOCKET_NETWORK socket2729 6568 driver39c
SOCKET_NETWORK socket2730 6569 driver39d

SOCKET_NETWORK socket2731 6570 driver40a
SOCKET_NETWORK socket2732 6571 driver40b
SOCKET_NETWORK socket2733 6572 driver40c
SOCKET_NETWORK socket2734 6573 driver40d
SOCKET_NETWORK socket2735 6574 driver40e
SOCKET_NETWORK socket2736 6575 driver40f
SOCKET_NETWORK socket2737 6576 driver40a
SOCKET_NETWORK socket2738 6577 driver40b
SOCKET_NETWORK socket2739 6578 driver40c
SOCKET_NETWORK socket2740 6579 driver40d
SOCKET_NETWORK socket2741 6580 driver40e
SOCKET_NETWORK socket2742 6581 driver40f
SOCKET_NETWORK socket2743 6582 driver40a
SOCKET_NETWORK socket2744 6583 driver40b
SOCKET_NETWORK socket2745 6584 driver40c
SOCKET_NETWORK socket2746 6585 driver40d
SOCKET_NETWORK socket2747 6586 driver40e
SOCKET_NETWORK socket2748 6587 driver40f
SOCKET_NETWORK socket2749 6588 driver40a
SOCKET_NETWORK socket2750 6589 driver40b
SOCKET_NETWORK socket2751 6590 driver40c
SOCKET_NETWORK socket2752 6591 driver40d
SOCKET_NETWORK socket2753 6592 driver40e
SOCKET_NETWORK socket2754 6593 driver40f
SOCKET_NETWORK socket2755 6594 driver40a
SOCKET_NETWORK socket2756 6595 driver40b
SOCKET_NETWORK socket2757 6596 driver40c
SOCKET_NETWORK socket2758 6597 driver40d
SOCKET_NETWORK socket2759 6598 driver40e
SOCKET_NETWORK socket2760 6599 driver40f
SOCKET_NETWORK socket2761 6600 driver40a
SOCKET_NETWORK socket2762 6601 driver40b
SOCKET_NETWORK socket2763 6602 driver40c
SOCKET_NETWORK socket2764 6603 driver40d
SOCKET_NETWORK socket2765 6604 driver40e
SOCKET_NETWORK socket2766 6605 driver40f
SOCKET_NETWORK socket2767 6606 driver40a
SOCKET_NETWORK socket2768 6607 driver40b
SOCKET_NETWORK socket2769 6608 driver40c
SOCKET_NETWORK socket2770 6609 driver40d
SOCKET_NETWORK socket2771 6610 driver40e
SOCKET_NETWORK socket2772 6611 driver40f
SOCKET_NETWORK socket2773 6612 driver40a
SOCKET_NETWORK socket2774 6613 driver40b
SOCKET_NETWORK socket2775 6614 driver40c
SOCKET_NETWORK socket2776 6615 driver40d
SOCKET_NETWORK socket2777 6616 driver40e
SOCKET_NETWORK socket2778 6617 driver40f
SOCKET_NETWORK socket2779 6618 driver40a
SOCKET_NETWORK socket2780 6619 driver40b
SOCKET_NETWORK socket2781 6620 driver40c
SOCKET_NETWORK socket2782 6621 driver40d
SOCKET_NETWORK socket2783 6622 driver40e
SOCKET_NETWORK socket2784 6623 driver40f
SOCKET_NETWORK socket2785 6624 driver40a
SOCKET_NETWORK socket2786 6625 driver40b
SOCKET_NETWORK socket2787 6626 driver40c
SOCKET_NETWORK socket2788 6627 driver40d
SOCKET_NETWORK socket2789 6628 driver40e
SOCKET_NETWORK socket2790 6629 driver40f
SOCKET_NETWORK socket2791 6630 driver40a
SOCKET_NETWORK socket2792 6631 driver40b
SOCKET_NETWORK socket2793 6632 driver40c
SOCKET_NETWORK socket2794 6633 driver40d
SOCKET_NETWORK socket2795 6634 driver40e
SOCKET_NETWORK socket2796 6635 driver40f
SOCKET_NETWORK socket2797 6636 driver40a
SOCKET_NETWORK socket2798 6637 driver40b
SOCKET_NETWORK socket2799 6638 driver40c
SOCKET_NETWORK socket2800 6639 driver40d
SOCKET_NETWORK socket2801 6500 driver41a
SOCKET_NETWORK socket2802 6501 driver41b
SOCKET_NETWORK socket2803 6502 driver41c
SOCKET_NETWORK socket2804 6503 driver41d
SOCKET_NETWORK socket2805 6504 driver41e
SOCKET_NETWORK socket2806 6505 driver41f
SOCKET_NETWORK socket2807 6506 driver41a
SOCKET_NETWORK socket2808 6507 driver41b
SOCKET_NETWORK socket2809 6508 driver41c
SOCKET_NETWORK socket2810 6509 driver41d
SOCKET_NETWORK socket2811 6510 driver41e
SOCKET_NETWORK socket2812 6511 driver41f
SOCKET_NETWORK socket2813 6512 driver41a
SOCKET_NETWORK socket2814 6513 driver41b
SOCKET_NETWORK socket2815 6514 driver41c
SOCKET_NETWORK socket2816 6515 driver41d
SOCKET_NETWORK socket2817 6516 driver41e
SOCKET_NETWORK socket2818 6517 driver41f
SOCKET_NETWORK socket2819 6518 driver41a
SOCKET_NETWORK socket2820 6519 driver41b
SOCKET_NETWORK socket2821 6520 driver41c
SOCKET_NETWORK socket2822 6521 driver41d
SOCKET_NETWORK socket2823 6522 driver41e
SOCKET_NETWORK socket2824 6523 driver41f
SOCKET_NETWORK socket2825 6524 driver41a
SOCKET_NETWORK socket2826 6525 driver41b
SOCKET_NETWORK socket2827 6526 driver41c
SOCKET_NETWORK socket2828 6527 driver41d
SOCKET_NETWORK socket2829 6528 driver41e
SOCKET_NETWORK socket2830 6529 driver41f
SOCKET_NETWORK socket2831 6530 driver41a
SOCKET_NETWORK socket2832 6531 driver41b
SOCKET_NETWORK socket2833 6532 driver41c
SOCKET_NETWORK socket2834 6533 driver41d
SOCKET_NETWORK socket2835 6534 driver41e
SOCKET_NETWORK socket2836 6535 driver41f
SOCKET_NETWORK socket2837 6536 driver41a
SOCKET_NETWORK socket2838 6537 driver41b
SOCKET_NETWORK socket2839 6538 driver41c
SOCKET_NETWORK socket2840 6539 driver41d
SOCKET_NETWORK socket2841 6540 driver41e
SOCKET_NETWORK socket2842 6541 driver41f
SOCKET_NETWORK socket2843 6542 driver41a
SOCKET_NETWORK socket2844 6543 driver41b
SOCKET_NETWORK socket2845 6544 driver41c
SOCKET_NETWORK socket2846 6545 driver41d
SOCKET_NETWORK socket2847 6546 driver41e
SOCKET_NETWORK socket2848 6547 driver41f
SOCKET_NETWORK socket2849 6548 driver41a
SOCKET_NETWORK socket2850 6549 driver41b
SOCKET_NETWORK socket2851 6550 driver41c
SOCKET_NETWORK socket2852 6551 driver41d
SOCKET_NETWORK socket2853 6552 driver41e
SOCKET_NETWORK socket2854 6553 driver41f
SOCKET_NETWORK socket2855 6554 driver41a
SOCKET_NETWORK socket2856 6555 driver41b
SOCKET_NETWORK socket2857 6556 driver41c
SOCKET_NETWORK socket2858 6557 driver41d
SOCKET_NETWORK socket2859 6558 driver41e
SOCKET_NETWORK socket2860 6559 driver41f
SOCKET_NETWORK socket2861 6560 driver41a
SOCKET_NETWORK socket2862 6561 driver41b
SOCKET_NETWORK socket2863 6562 driver41c
SOCKET_NETWORK socket2864 6563 driver41d
SOCKET_NETWORK socket2865 6564 driver41e
SOCKET_NETWORK socket2866 6565 driver41f
SOCKET_NETWORK socket2867 6566 driver41a
SOCKET_NETWORK socket2868 6567 driver41b
SOCKET_NETWORK socket2869 6568 driver41c
SOCKET_NETWORK socket2870 6569 driver41d
SOCKET_NETWORK socket2871 6570 driver42a
SOCKET_NETWORK socket2872 6571 driver42b
SOCKET_NETWORK socket2873 6572 driver42c
SOCKET_NETWORK socket2874 6573 driver42d
SOCKET_NETWORK socket2875 6574 driver42e
SOCKET_NETWORK socket2876 6575 driver42f
SOCKET_NETWORK socket2877 6576 driver42a
SOCKET_NETWORK socket2878 6577 driver42b
SOCKET_NETWORK socket2879 6578 driver42c
SOCKET_NETWORK socket2880 6579 driver42d
SOCKET_NETWORK socket2881 6580 driver42e
SOCKET_NETWORK socket2882 6581 driver42f
SOCKET_NETWORK socket2883 6582 driver42a
SOCKET_NETWORK socket2884 6583 driver42b
SOCKET_NETWORK socket2885 6584 driver42c
SOCKET_NETWORK socket2886 6585 driver42d
SOCKET_NETWORK socket2887 6586 driver42e
SOCKET_NETWORK socket2888 6587 driver42f
SOCKET_NETWORK socket2889 6588 driver42a
SOCKET_NETWORK socket2890 6589 driver42b
SOCKET_NETWORK socket2891 6590 driver42c
SOCKET_NETWORK socket2892 6591 driver42d
SOCKET_NETWORK socket2893 6592 driver42e
SOCKET_NETWORK socket2894 6593 driver42f
SOCKET_NETWORK socket2895 6594 driver42a
SOCKET_NETWORK socket2896 6595 driver42b
SOCKET_NETWORK socket2897 6596 driver42c
SOCKET_NETWORK socket2898 6597 driver42d
SOCKET_NETWORK socket2899 6598 driver42e
SOCKET_NETWORK socket2900 6599 driver42f
SOCKET_NETWORK socket2901 6600 driver42a
SOCKET_NETWORK socket2902 6601 driver42b
SOCKET_NETWORK socket2903 6602 driver42c
SOCKET_NETWORK socket2904 6603 driver42d
SOCKET_NETWORK socket2905 6604 driver42e
SOCKET_NETWORK socket2906 6605 driver42f
SOCKET_NETWORK socket2907 6606 driver42a
SOCKET_NETWORK socket2908 6607 driver42b
SOCKET_NETWORK socket2909 6608 driver42c
SOCKET_NETWORK socket2910 6609 driver42d
SOCKET_NETWORK socket2911 6610 driver42e
SOCKET_NETWORK socket2912 6611 driver42f
SOCKET_NETWORK socket2913 6612 driver42a
SOCKET_NETWORK socket2914 6613 driver42b
SOCKET_NETWORK socket2915 6614 driver42c
SOCKET_NETWORK socket2916 6615 driver42d
SOCKET_NETWORK socket2917 6616 driver42e
SOCKET_NETWORK socket2918 6617 driver42f
SOCKET_NETWORK socket2919 6618 driver42a
SOCKET_NETWORK socket2920 6619 driver42b
SOCKET_NETWORK socket2921 6620 driver42c
SOCKET_NETWORK socket2922 6621 driver42d
SOCKET_NETWORK socket2923 6622 driver42e
SOCKET_NETWORK socket2924 6623 driver42f
SOCKET_NETWORK socket2925 6624 driver42a
SOCKET_NETWORK socket2926 6625 driver42b
SOCKET_NETWORK socket2927 6626 driver42c
SOCKET_NETWORK socket2928 6627 driver42d

```
SOCKET_NETWORK socket2929  6628 driver42e
SOCKET_NETWORK socket2930  6629 driver42f
SOCKET_NETWORK socket2931  6630 driver42a
SOCKET_NETWORK socket2932  6631 driver42b
SOCKET_NETWORK socket2933  6632 driver42c
SOCKET_NETWORK socket2934  6633 driver42d
SOCKET_NETWORK socket2935  6634 driver42e
SOCKET_NETWORK socket2936  6635 driver42f
SOCKET_NETWORK socket2937  6636 driver42a
SOCKET_NETWORK socket2938  6637 driver42b
SOCKET_NETWORK socket2939  6638 driver42c
SOCKET_NETWORK socket2940  6639 driver42d
#elif MASTER_NUM8
SOCKET_NETWORK socket2941  6500 driver43a
SOCKET_NETWORK socket2942  6501 driver43b
SOCKET_NETWORK socket2943  6502 driver43c
SOCKET_NETWORK socket2944  6503 driver43d
SOCKET_NETWORK socket2945  6504 driver43e
SOCKET_NETWORK socket2946  6505 driver43f
SOCKET_NETWORK socket2947  6506 driver43a
SOCKET_NETWORK socket2948  6507 driver43b
SOCKET_NETWORK socket2949  6508 driver43c
SOCKET_NETWORK socket2950  6509 driver43d
SOCKET_NETWORK socket2951  6510 driver43e
SOCKET_NETWORK socket2952  6511 driver43f
SOCKET_NETWORK socket2953  6512 driver43a
SOCKET_NETWORK socket2954  6513 driver43b
SOCKET_NETWORK socket2955  6514 driver43c
SOCKET_NETWORK socket2956  6515 driver43d
SOCKET_NETWORK socket2957  6516 driver43e
SOCKET_NETWORK socket2958  6517 driver43f
SOCKET_NETWORK socket2959  6518 driver43a
SOCKET_NETWORK socket2960  6519 driver43b
SOCKET_NETWORK socket2961  6520 driver43c
SOCKET_NETWORK socket2962  6521 driver43d
SOCKET_NETWORK socket2963  6522 driver43e
SOCKET_NETWORK socket2964  6523 driver43f
SOCKET_NETWORK socket2965  6524 driver43a
SOCKET_NETWORK socket2966  6525 driver43b
SOCKET_NETWORK socket2967  6526 driver43c
SOCKET_NETWORK socket2968  6527 driver43d
SOCKET_NETWORK socket2969  6528 driver43e
SOCKET_NETWORK socket2970  6529 driver43f
SOCKET_NETWORK socket2971  6530 driver43a
SOCKET_NETWORK socket2972  6531 driver43b
SOCKET_NETWORK socket2973  6532 driver43c
SOCKET_NETWORK socket2974  6533 driver43d
SOCKET_NETWORK socket2975  6534 driver43e
SOCKET_NETWORK socket2976  6535 driver43f
SOCKET_NETWORK socket2977  6536 driver43a
SOCKET_NETWORK socket2978  6537 driver43b
SOCKET_NETWORK socket2979  6538 driver43c
SOCKET_NETWORK socket2980  6539 driver43d
SOCKET_NETWORK socket2981  6540 driver43e
SOCKET_NETWORK socket2982  6541 driver43f
SOCKET_NETWORK socket2983  6542 driver43a
SOCKET_NETWORK socket2984  6543 driver43b
SOCKET_NETWORK socket2985  6544 driver43c
SOCKET_NETWORK socket2986  6545 driver43d
SOCKET_NETWORK socket2987  6546 driver43e
SOCKET_NETWORK socket2988  6547 driver43f
SOCKET_NETWORK socket2989  6548 driver43a
SOCKET_NETWORK socket2990  6549 driver43b
SOCKET_NETWORK socket2991  6550 driver43c
SOCKET_NETWORK socket2992  6551 driver43d
SOCKET_NETWORK socket2993  6552 driver43e
SOCKET_NETWORK socket2994  6553 driver43f
SOCKET_NETWORK socket2995  6554 driver43a
SOCKET_NETWORK socket2996  6555 driver43b
SOCKET_NETWORK socket2997  6556 driver43c
SOCKET_NETWORK socket2998  6557 driver43d
SOCKET_NETWORK socket2999  6558 driver43e
SOCKET_NETWORK socket3000  6559 driver43f
SOCKET_NETWORK socket3001  6560 driver43a
SOCKET_NETWORK socket3002  6561 driver43b
SOCKET_NETWORK socket3003  6562 driver43c
SOCKET_NETWORK socket3004  6563 driver43d
SOCKET_NETWORK socket3005  6564 driver43e
SOCKET_NETWORK socket3006  6565 driver43f
SOCKET_NETWORK socket3007  6566 driver43a
SOCKET_NETWORK socket3008  6567 driver43b
SOCKET_NETWORK socket3009  6568 driver43c
SOCKET_NETWORK socket3010  6569 driver43d
SOCKET_NETWORK socket3011  6570 driver44a
SOCKET_NETWORK socket3012  6571 driver44b
SOCKET_NETWORK socket3013  6572 driver44c
SOCKET_NETWORK socket3014  6573 driver44d
SOCKET_NETWORK socket3015  6574 driver44e
SOCKET_NETWORK socket3016  6575 driver44f
SOCKET_NETWORK socket3017  6576 driver44a
SOCKET_NETWORK socket3018  6577 driver44b
SOCKET_NETWORK socket3019  6578 driver44c
SOCKET_NETWORK socket3020  6579 driver44d
SOCKET_NETWORK socket3021  6580 driver44e
SOCKET_NETWORK socket3022  6581 driver44f
SOCKET_NETWORK socket3023  6582 driver44a
SOCKET_NETWORK socket3024  6583 driver44b
SOCKET_NETWORK socket3025  6584 driver44c
SOCKET_NETWORK socket3026  6585 driver44d
SOCKET_NETWORK socket3027  6586 driver44e
SOCKET_NETWORK socket3028  6587 driver44f
SOCKET_NETWORK socket3029  6588 driver44a
SOCKET_NETWORK socket3030  6589 driver44b
SOCKET_NETWORK socket3031  6590 driver44c
SOCKET_NETWORK socket3032  6591 driver44d
SOCKET_NETWORK socket3033  6592 driver44e
SOCKET_NETWORK socket3034  6593 driver44f
SOCKET_NETWORK socket3035  6594 driver44a
SOCKET_NETWORK socket3036  6595 driver44b
SOCKET_NETWORK socket3037  6596 driver44c
SOCKET_NETWORK socket3038  6597 driver44d
SOCKET_NETWORK socket3039  6598 driver44e
SOCKET_NETWORK socket3040  6599 driver44f
SOCKET_NETWORK socket3041  6600 driver44a
SOCKET_NETWORK socket3042  6601 driver44b
SOCKET_NETWORK socket3043  6602 driver44c
SOCKET_NETWORK socket3044  6603 driver44d
SOCKET_NETWORK socket3045  6604 driver44e
SOCKET_NETWORK socket3046  6605 driver44f
SOCKET_NETWORK socket3047  6606 driver44a
SOCKET_NETWORK socket3048  6607 driver44b
SOCKET_NETWORK socket3049  6608 driver44c
SOCKET_NETWORK socket3050  6609 driver44d
SOCKET_NETWORK socket3051  6610 driver44e
SOCKET_NETWORK socket3052  6611 driver44f
SOCKET_NETWORK socket3053  6612 driver44a
SOCKET_NETWORK socket3054  6613 driver44b
SOCKET_NETWORK socket3055  6614 driver44c
SOCKET_NETWORK socket3056  6615 driver44d
SOCKET_NETWORK socket3057  6616 driver44e
SOCKET_NETWORK socket3058  6617 driver44f
SOCKET_NETWORK socket3059  6618 driver44a
SOCKET_NETWORK socket3060  6619 driver44b
SOCKET_NETWORK socket3061  6620 driver44c
SOCKET_NETWORK socket3062  6621 driver44d
SOCKET_NETWORK socket3063  6622 driver44e
SOCKET_NETWORK socket3064  6623 driver44f
SOCKET_NETWORK socket3065  6624 driver44a
SOCKET_NETWORK socket3066  6625 driver44b
SOCKET_NETWORK socket3067  6626 driver44c
SOCKET_NETWORK socket3068  6627 driver44d
SOCKET_NETWORK socket3069  6628 driver44e
SOCKET_NETWORK socket3070  6629 driver44f
SOCKET_NETWORK socket3071  6630 driver44a
SOCKET_NETWORK socket3072  6631 driver44b
SOCKET_NETWORK socket3073  6632 driver44c
SOCKET_NETWORK socket3074  6633 driver44d
SOCKET_NETWORK socket3075  6634 driver44e
SOCKET_NETWORK socket3076  6635 driver44f
SOCKET_NETWORK socket3077  6636 driver44a
SOCKET_NETWORK socket3078  6637 driver44b
SOCKET_NETWORK socket3079  6638 driver44c
SOCKET_NETWORK socket3080  6639 driver44d
SOCKET_NETWORK socket3081  6500 driver45a
SOCKET_NETWORK socket3082  6501 driver45b
SOCKET_NETWORK socket3083  6502 driver45c
SOCKET_NETWORK socket3084  6503 driver45d
SOCKET_NETWORK socket3085  6504 driver45e
SOCKET_NETWORK socket3086  6505 driver45f
SOCKET_NETWORK socket3087  6506 driver45a
SOCKET_NETWORK socket3088  6507 driver45b
SOCKET_NETWORK socket3089  6508 driver45c
SOCKET_NETWORK socket3090  6509 driver45d
SOCKET_NETWORK socket3091  6510 driver45e
SOCKET_NETWORK socket3092  6511 driver45f
SOCKET_NETWORK socket3093  6512 driver45a
SOCKET_NETWORK socket3094  6513 driver45b
SOCKET_NETWORK socket3095  6514 driver45c
SOCKET_NETWORK socket3096  6515 driver45d
SOCKET_NETWORK socket3097  6516 driver45e
SOCKET_NETWORK socket3098  6517 driver45f
SOCKET_NETWORK socket3099  6518 driver45a
SOCKET_NETWORK socket3100  6519 driver45b
SOCKET_NETWORK socket3101  6520 driver45c
SOCKET_NETWORK socket3102  6521 driver45d
SOCKET_NETWORK socket3103  6522 driver45e
SOCKET_NETWORK socket3104  6523 driver45f
SOCKET_NETWORK socket3105  6524 driver45a
SOCKET_NETWORK socket3106  6525 driver45b
SOCKET_NETWORK socket3107  6526 driver45c
SOCKET_NETWORK socket3108  6527 driver45d
SOCKET_NETWORK socket3109  6528 driver45e
SOCKET_NETWORK socket3110  6529 driver45f
SOCKET_NETWORK socket3111  6530 driver45a
SOCKET_NETWORK socket3112  6531 driver45b
SOCKET_NETWORK socket3113  6532 driver45c
SOCKET_NETWORK socket3114  6533 driver45d
SOCKET_NETWORK socket3115  6534 driver45e
SOCKET_NETWORK socket3116  6535 driver45f
SOCKET_NETWORK socket3117  6536 driver45a
SOCKET_NETWORK socket3118  6537 driver45b
SOCKET_NETWORK socket3119  6538 driver45c
SOCKET_NETWORK socket3120  6539 driver45d
SOCKET_NETWORK socket3121  6540 driver45e
SOCKET_NETWORK socket3122  6541 driver45f
SOCKET_NETWORK socket3123  6542 driver45a
SOCKET_NETWORK socket3124  6543 driver45b
SOCKET_NETWORK socket3125  6544 driver45c
```

SOCKET_NETWORK socket3126 6545 driver45d
SOCKET_NETWORK socket3127 6546 driver45e
SOCKET_NETWORK socket3128 6547 driver45f
SOCKET_NETWORK socket3129 6548 driver45a
SOCKET_NETWORK socket3130 6549 driver45b
SOCKET_NETWORK socket3131 6550 driver45c
SOCKET_NETWORK socket3132 6551 driver45d
SOCKET_NETWORK socket3133 6552 driver45e
SOCKET_NETWORK socket3134 6553 driver45f
SOCKET_NETWORK socket3135 6554 driver45a
SOCKET_NETWORK socket3136 6555 driver45b
SOCKET_NETWORK socket3137 6556 driver45c
SOCKET_NETWORK socket3138 6557 driver45d
SOCKET_NETWORK socket3139 6558 driver45e
SOCKET_NETWORK socket3140 6559 driver45f
SOCKET_NETWORK socket3141 6560 driver45a
SOCKET_NETWORK socket3142 6561 driver45b
SOCKET_NETWORK socket3143 6562 driver45c
SOCKET_NETWORK socket3144 6563 driver45d
SOCKET_NETWORK socket3145 6564 driver45e
SOCKET_NETWORK socket3146 6565 driver45f
SOCKET_NETWORK socket3147 6566 driver45a
SOCKET_NETWORK socket3148 6567 driver45b
SOCKET_NETWORK socket3149 6568 driver45c
SOCKET_NETWORK socket3150 6569 driver45d
SOCKET_NETWORK socket3151 6570 driver46a
SOCKET_NETWORK socket3152 6571 driver46b
SOCKET_NETWORK socket3153 6572 driver46c
SOCKET_NETWORK socket3154 6573 driver46d
SOCKET_NETWORK socket3155 6574 driver46e
SOCKET_NETWORK socket3156 6575 driver46f
SOCKET_NETWORK socket3157 6576 driver46a
SOCKET_NETWORK socket3158 6577 driver46b
SOCKET_NETWORK socket3159 6578 driver46c
SOCKET_NETWORK socket3160 6579 driver46d
SOCKET_NETWORK socket3161 6580 driver46e
SOCKET_NETWORK socket3162 6581 driver46f
SOCKET_NETWORK socket3163 6582 driver46a
SOCKET_NETWORK socket3164 6583 driver46b
SOCKET_NETWORK socket3165 6584 driver46c
SOCKET_NETWORK socket3166 6585 driver46d
SOCKET_NETWORK socket3167 6586 driver46e
SOCKET_NETWORK socket3168 6587 driver46f
SOCKET_NETWORK socket3169 6588 driver46a
SOCKET_NETWORK socket3170 6589 driver46b
SOCKET_NETWORK socket3171 6590 driver46c
SOCKET_NETWORK socket3172 6591 driver46d
SOCKET_NETWORK socket3173 6592 driver46e
SOCKET_NETWORK socket3174 6593 driver46f
SOCKET_NETWORK socket3175 6594 driver46a
SOCKET_NETWORK socket3176 6595 driver46b
SOCKET_NETWORK socket3177 6596 driver46c
SOCKET_NETWORK socket3178 6597 driver46d
SOCKET_NETWORK socket3179 6598 driver46e
SOCKET_NETWORK socket3180 6599 driver46f
SOCKET_NETWORK socket3181 6600 driver46a
SOCKET_NETWORK socket3182 6601 driver46b
SOCKET_NETWORK socket3183 6602 driver46c
SOCKET_NETWORK socket3184 6603 driver46d
SOCKET_NETWORK socket3185 6604 driver46e
SOCKET_NETWORK socket3186 6605 driver46f
SOCKET_NETWORK socket3187 6606 driver46a
SOCKET_NETWORK socket3188 6607 driver46b
SOCKET_NETWORK socket3189 6608 driver46c
SOCKET_NETWORK socket3190 6609 driver46d
SOCKET_NETWORK socket3191 6610 driver46e
SOCKET_NETWORK socket3192 6611 driver46f
SOCKET_NETWORK socket3193 6612 driver46a
SOCKET_NETWORK socket3194 6613 driver46b
SOCKET_NETWORK socket3195 6614 driver46c
SOCKET_NETWORK socket3196 6615 driver46d
SOCKET_NETWORK socket3197 6616 driver46e
SOCKET_NETWORK socket3198 6617 driver46f
SOCKET_NETWORK socket3199 6618 driver46a
SOCKET_NETWORK socket3200 6619 driver46b
SOCKET_NETWORK socket3201 6620 driver46c
SOCKET_NETWORK socket3202 6621 driver46d
SOCKET_NETWORK socket3203 6622 driver46e
SOCKET_NETWORK socket3204 6623 driver46f
SOCKET_NETWORK socket3205 6624 driver46a
SOCKET_NETWORK socket3206 6625 driver46b
SOCKET_NETWORK socket3207 6626 driver46c
SOCKET_NETWORK socket3208 6627 driver46d
SOCKET_NETWORK socket3209 6628 driver46e
SOCKET_NETWORK socket3210 6629 driver46f
SOCKET_NETWORK socket3211 6630 driver46a
SOCKET_NETWORK socket3212 6631 driver46b
SOCKET_NETWORK socket3213 6632 driver46c
SOCKET_NETWORK socket3214 6633 driver46d
SOCKET_NETWORK socket3215 6634 driver46e
SOCKET_NETWORK socket3216 6635 driver46f
SOCKET_NETWORK socket3217 6636 driver46a
SOCKET_NETWORK socket3218 6637 driver46b
SOCKET_NETWORK socket3219 6638 driver46c
SOCKET_NETWORK socket3220 6639 driver46d
SOCKET_NETWORK socket3221 6500 driver47a
SOCKET_NETWORK socket3222 6501 driver47b
SOCKET_NETWORK socket3223 6502 driver47c
SOCKET_NETWORK socket3224 6503 driver47d
SOCKET_NETWORK socket3225 6504 driver47e
SOCKET_NETWORK socket3226 6505 driver47f
SOCKET_NETWORK socket3227 6506 driver47a
SOCKET_NETWORK socket3228 6507 driver47b
SOCKET_NETWORK socket3229 6508 driver47c
SOCKET_NETWORK socket3230 6509 driver47d
SOCKET_NETWORK socket3231 6510 driver47e
SOCKET_NETWORK socket3232 6511 driver47f
SOCKET_NETWORK socket3233 6512 driver47a
SOCKET_NETWORK socket3234 6513 driver47b
SOCKET_NETWORK socket3235 6514 driver47c
SOCKET_NETWORK socket3236 6515 driver47d
SOCKET_NETWORK socket3237 6516 driver47e
SOCKET_NETWORK socket3238 6517 driver47f
SOCKET_NETWORK socket3239 6518 driver47a
SOCKET_NETWORK socket3240 6519 driver47b
SOCKET_NETWORK socket3241 6520 driver47c
SOCKET_NETWORK socket3242 6521 driver47d
SOCKET_NETWORK socket3243 6522 driver47e
SOCKET_NETWORK socket3244 6523 driver47f
SOCKET_NETWORK socket3245 6524 driver47a
SOCKET_NETWORK socket3246 6525 driver47b
SOCKET_NETWORK socket3247 6526 driver47c
SOCKET_NETWORK socket3248 6527 driver47d
SOCKET_NETWORK socket3249 6528 driver47e
SOCKET_NETWORK socket3250 6529 driver47f
SOCKET_NETWORK socket3251 6530 driver47a
SOCKET_NETWORK socket3252 6531 driver47b
SOCKET_NETWORK socket3253 6532 driver47c
SOCKET_NETWORK socket3254 6533 driver47d
SOCKET_NETWORK socket3255 6534 driver47e
SOCKET_NETWORK socket3256 6535 driver47f
SOCKET_NETWORK socket3257 6536 driver47a
SOCKET_NETWORK socket3258 6537 driver47b
SOCKET_NETWORK socket3259 6538 driver47c
SOCKET_NETWORK socket3260 6539 driver47d
SOCKET_NETWORK socket3261 6540 driver47e
SOCKET_NETWORK socket3262 6541 driver47f
SOCKET_NETWORK socket3263 6542 driver47a
SOCKET_NETWORK socket3264 6543 driver47b
SOCKET_NETWORK socket3265 6544 driver47c
SOCKET_NETWORK socket3266 6545 driver47d
SOCKET_NETWORK socket3267 6546 driver47e
SOCKET_NETWORK socket3268 6547 driver47f
SOCKET_NETWORK socket3269 6548 driver47a
SOCKET_NETWORK socket3270 6549 driver47b
SOCKET_NETWORK socket3271 6550 driver47c
SOCKET_NETWORK socket3272 6551 driver47d
SOCKET_NETWORK socket3273 6552 driver47e
SOCKET_NETWORK socket3274 6553 driver47f
SOCKET_NETWORK socket3275 6554 driver47a
SOCKET_NETWORK socket3276 6555 driver47b
SOCKET_NETWORK socket3277 6556 driver47c
SOCKET_NETWORK socket3278 6557 driver47d
SOCKET_NETWORK socket3279 6558 driver47e
SOCKET_NETWORK socket3280 6559 driver47f
SOCKET_NETWORK socket3281 6560 driver47a
SOCKET_NETWORK socket3282 6561 driver47b
SOCKET_NETWORK socket3283 6562 driver47c
SOCKET_NETWORK socket3284 6563 driver47d
SOCKET_NETWORK socket3285 6564 driver47e
SOCKET_NETWORK socket3286 6565 driver47f
SOCKET_NETWORK socket3287 6566 driver47a
SOCKET_NETWORK socket3288 6567 driver47b
SOCKET_NETWORK socket3289 6568 driver47c
SOCKET_NETWORK socket3290 6569 driver47d
SOCKET_NETWORK socket3291 6570 driver48a
SOCKET_NETWORK socket3292 6571 driver48b
SOCKET_NETWORK socket3293 6572 driver48c
SOCKET_NETWORK socket3294 6573 driver48d
SOCKET_NETWORK socket3295 6574 driver48e
SOCKET_NETWORK socket3296 6575 driver48f
SOCKET_NETWORK socket3297 6576 driver48a
SOCKET_NETWORK socket3298 6577 driver48b
SOCKET_NETWORK socket3299 6578 driver48c
SOCKET_NETWORK socket3300 6579 driver48d
SOCKET_NETWORK socket3301 6580 driver48e
SOCKET_NETWORK socket3302 6581 driver48f
SOCKET_NETWORK socket3303 6582 driver48a
SOCKET_NETWORK socket3304 6583 driver48b
SOCKET_NETWORK socket3305 6584 driver48c
SOCKET_NETWORK socket3306 6585 driver48d
SOCKET_NETWORK socket3307 6586 driver48e
SOCKET_NETWORK socket3308 6587 driver48f
SOCKET_NETWORK socket3309 6588 driver48a
SOCKET_NETWORK socket3310 6589 driver48b
SOCKET_NETWORK socket3311 6590 driver48c
SOCKET_NETWORK socket3312 6591 driver48d
SOCKET_NETWORK socket3313 6592 driver48e
SOCKET_NETWORK socket3314 6593 driver48f
SOCKET_NETWORK socket3315 6594 driver48a
SOCKET_NETWORK socket3316 6595 driver48b
SOCKET_NETWORK socket3317 6596 driver48c
SOCKET_NETWORK socket3318 6597 driver48d
SOCKET_NETWORK socket3319 6598 driver48e
SOCKET_NETWORK socket3320 6599 driver48f
SOCKET_NETWORK socket3321 6600 driver48a
SOCKET_NETWORK socket3322 6601 driver48b
SOCKET_NETWORK socket3323 6602 driver48c

```
SOCKET_NETWORK socket3324  6603 driver48d
SOCKET_NETWORK socket3325  6604 driver48e
SOCKET_NETWORK socket3326  6605 driver48f
SOCKET_NETWORK socket3327  6606 driver48a
SOCKET_NETWORK socket3328  6607 driver48b
SOCKET_NETWORK socket3329  6608 driver48c
SOCKET_NETWORK socket3330  6609 driver48d
SOCKET_NETWORK socket3331  6610 driver48e
SOCKET_NETWORK socket3332  6611 driver48f
SOCKET_NETWORK socket3333  6612 driver48a
SOCKET_NETWORK socket3334  6613 driver48b
SOCKET_NETWORK socket3335  6614 driver48c
SOCKET_NETWORK socket3336  6615 driver48d
SOCKET_NETWORK socket3337  6616 driver48e
SOCKET_NETWORK socket3338  6617 driver48f
SOCKET_NETWORK socket3339  6618 driver48a
SOCKET_NETWORK socket3340  6619 driver48b
SOCKET_NETWORK socket3341  6620 driver48c
SOCKET_NETWORK socket3342  6621 driver48d
SOCKET_NETWORK socket3343  6622 driver48e
SOCKET_NETWORK socket3344  6623 driver48f
SOCKET_NETWORK socket3345  6624 driver48a
SOCKET_NETWORK socket3346  6625 driver48b
SOCKET_NETWORK socket3347  6626 driver48c
SOCKET_NETWORK socket3348  6627 driver48d
SOCKET_NETWORK socket3349  6628 driver48e
SOCKET_NETWORK socket3350  6629 driver48f
SOCKET_NETWORK socket3351  6630 driver48a
SOCKET_NETWORK socket3352  6631 driver48b
SOCKET_NETWORK socket3353  6632 driver48c
SOCKET_NETWORK socket3354  6633 driver48d
SOCKET_NETWORK socket3355  6634 driver48e
SOCKET_NETWORK socket3356  6635 driver48f
SOCKET_NETWORK socket3357  6636 driver48a
SOCKET_NETWORK socket3358  6637 driver48b
SOCKET_NETWORK socket3359  6638 driver48c
SOCKET_NETWORK socket3360  6639 driver48d
#elif MASTER_NUM9
SOCKET_NETWORK socket3361  6500 driver49a
SOCKET_NETWORK socket3362  6501 driver49b
SOCKET_NETWORK socket3363  6502 driver49c
SOCKET_NETWORK socket3364  6503 driver49d
SOCKET_NETWORK socket3365  6504 driver49e
SOCKET_NETWORK socket3366  6505 driver49f
SOCKET_NETWORK socket3367  6506 driver49a
SOCKET_NETWORK socket3368  6507 driver49b
SOCKET_NETWORK socket3369  6508 driver49c
SOCKET_NETWORK socket3370  6509 driver49d
SOCKET_NETWORK socket3371  6510 driver49e
SOCKET_NETWORK socket3372  6511 driver49f
SOCKET_NETWORK socket3373  6512 driver49a
SOCKET_NETWORK socket3374  6513 driver49b
SOCKET_NETWORK socket3375  6514 driver49c
SOCKET_NETWORK socket3376  6515 driver49d
SOCKET_NETWORK socket3377  6516 driver49e
SOCKET_NETWORK socket3378  6517 driver49f
SOCKET_NETWORK socket3379  6518 driver49a
SOCKET_NETWORK socket3380  6519 driver49b
SOCKET_NETWORK socket3381  6520 driver49c
SOCKET_NETWORK socket3382  6521 driver49d
SOCKET_NETWORK socket3383  6522 driver49e
SOCKET_NETWORK socket3384  6523 driver49f
SOCKET_NETWORK socket3385  6524 driver49a
SOCKET_NETWORK socket3386  6525 driver49b
SOCKET_NETWORK socket3387  6526 driver49c
SOCKET_NETWORK socket3388  6527 driver49d
SOCKET_NETWORK socket3389  6528 driver49e
SOCKET_NETWORK socket3390  6529 driver49f
SOCKET_NETWORK socket3391  6530 driver49a
SOCKET_NETWORK socket3392  6531 driver49b
SOCKET_NETWORK socket3393  6532 driver49c
SOCKET_NETWORK socket3394  6533 driver49d
SOCKET_NETWORK socket3395  6534 driver49e
SOCKET_NETWORK socket3396  6535 driver49f
SOCKET_NETWORK socket3397  6536 driver49a
SOCKET_NETWORK socket3398  6537 driver49b
SOCKET_NETWORK socket3399  6538 driver49c
SOCKET_NETWORK socket3400  6539 driver49d
SOCKET_NETWORK socket3401  6540 driver49e
SOCKET_NETWORK socket3402  6541 driver49f
SOCKET_NETWORK socket3403  6542 driver49a
SOCKET_NETWORK socket3404  6543 driver49b
SOCKET_NETWORK socket3405  6544 driver49c
SOCKET_NETWORK socket3406  6545 driver49d
SOCKET_NETWORK socket3407  6546 driver49e
SOCKET_NETWORK socket3408  6547 driver49f
SOCKET_NETWORK socket3409  6548 driver49a
SOCKET_NETWORK socket3410  6549 driver49b
SOCKET_NETWORK socket3411  6550 driver49c
SOCKET_NETWORK socket3412  6551 driver49d
SOCKET_NETWORK socket3413  6552 driver49e
SOCKET_NETWORK socket3414  6553 driver49f
SOCKET_NETWORK socket3415  6554 driver49a
SOCKET_NETWORK socket3416  6555 driver49b
SOCKET_NETWORK socket3417  6556 driver49c
SOCKET_NETWORK socket3418  6557 driver49d
SOCKET_NETWORK socket3419  6558 driver49e
SOCKET_NETWORK socket3420  6559 driver49f
SOCKET_NETWORK socket3421  6560 driver49a
SOCKET_NETWORK socket3422  6561 driver49b
SOCKET_NETWORK socket3423  6562 driver49c
SOCKET_NETWORK socket3424  6563 driver49d
SOCKET_NETWORK socket3425  6564 driver49e
SOCKET_NETWORK socket3426  6565 driver49f
SOCKET_NETWORK socket3427  6566 driver49a
SOCKET_NETWORK socket3428  6567 driver49b
SOCKET_NETWORK socket3429  6568 driver49c
SOCKET_NETWORK socket3430  6569 driver49d
SOCKET_NETWORK socket3431  6570 driver50a
SOCKET_NETWORK socket3432  6571 driver50b
SOCKET_NETWORK socket3433  6572 driver50c
SOCKET_NETWORK socket3434  6573 driver50d
SOCKET_NETWORK socket3435  6574 driver50e
SOCKET_NETWORK socket3436  6575 driver50f
SOCKET_NETWORK socket3437  6576 driver50a
SOCKET_NETWORK socket3438  6577 driver50b
SOCKET_NETWORK socket3439  6578 driver50c
SOCKET_NETWORK socket3440  6579 driver50d
SOCKET_NETWORK socket3441  6580 driver50e
SOCKET_NETWORK socket3442  6581 driver50f
SOCKET_NETWORK socket3443  6582 driver50a
SOCKET_NETWORK socket3444  6583 driver50b
SOCKET_NETWORK socket3445  6584 driver50c
SOCKET_NETWORK socket3446  6585 driver50d
SOCKET_NETWORK socket3447  6586 driver50e
SOCKET_NETWORK socket3448  6587 driver50f
SOCKET_NETWORK socket3449  6588 driver50a
SOCKET_NETWORK socket3450  6589 driver50b
SOCKET_NETWORK socket3451  6590 driver50c
SOCKET_NETWORK socket3452  6591 driver50d
SOCKET_NETWORK socket3453  6592 driver50e
SOCKET_NETWORK socket3454  6593 driver50f
SOCKET_NETWORK socket3455  6594 driver50a
SOCKET_NETWORK socket3456  6595 driver50b
SOCKET_NETWORK socket3457  6596 driver50c
SOCKET_NETWORK socket3458  6597 driver50d
SOCKET_NETWORK socket3459  6598 driver50e
SOCKET_NETWORK socket3460  6599 driver50f
SOCKET_NETWORK socket3461  6600 driver50a
SOCKET_NETWORK socket3462  6601 driver50b
SOCKET_NETWORK socket3463  6602 driver50c
SOCKET_NETWORK socket3464  6603 driver50d
SOCKET_NETWORK socket3465  6604 driver50e
SOCKET_NETWORK socket3466  6605 driver50f
SOCKET_NETWORK socket3467  6606 driver50a
SOCKET_NETWORK socket3468  6607 driver50b
SOCKET_NETWORK socket3469  6608 driver50c
SOCKET_NETWORK socket3470  6609 driver50d
SOCKET_NETWORK socket3471  6610 driver50e
SOCKET_NETWORK socket3472  6611 driver50f
SOCKET_NETWORK socket3473  6612 driver50a
SOCKET_NETWORK socket3474  6613 driver50b
SOCKET_NETWORK socket3475  6614 driver50c
SOCKET_NETWORK socket3476  6615 driver50d
SOCKET_NETWORK socket3477  6616 driver50e
SOCKET_NETWORK socket3478  6617 driver50f
SOCKET_NETWORK socket3479  6618 driver50a
SOCKET_NETWORK socket3480  6619 driver50b
SOCKET_NETWORK socket3481  6620 driver50c
SOCKET_NETWORK socket3482  6621 driver50d
SOCKET_NETWORK socket3483  6622 driver50e
SOCKET_NETWORK socket3484  6623 driver50f
SOCKET_NETWORK socket3485  6624 driver50a
SOCKET_NETWORK socket3486  6625 driver50b
SOCKET_NETWORK socket3487  6626 driver50c
SOCKET_NETWORK socket3488  6627 driver50d
SOCKET_NETWORK socket3489  6628 driver50e
SOCKET_NETWORK socket3490  6629 driver50f
SOCKET_NETWORK socket3491  6630 driver50a
SOCKET_NETWORK socket3492  6631 driver50b
SOCKET_NETWORK socket3493  6632 driver50c
SOCKET_NETWORK socket3494  6633 driver50d
SOCKET_NETWORK socket3495  6634 driver50e
SOCKET_NETWORK socket3496  6635 driver50f
SOCKET_NETWORK socket3497  6636 driver50a
SOCKET_NETWORK socket3498  6637 driver50b
SOCKET_NETWORK socket3499  6638 driver50c
SOCKET_NETWORK socket3500  6639 driver50d
SOCKET_NETWORK socket3501  6500 driver51a
SOCKET_NETWORK socket3502  6501 driver51b
SOCKET_NETWORK socket3503  6502 driver51c
SOCKET_NETWORK socket3504  6503 driver51d
SOCKET_NETWORK socket3505  6504 driver51e
SOCKET_NETWORK socket3506  6505 driver51f
SOCKET_NETWORK socket3507  6506 driver51a
SOCKET_NETWORK socket3508  6507 driver51b
SOCKET_NETWORK socket3509  6508 driver51c
SOCKET_NETWORK socket3510  6509 driver51d
SOCKET_NETWORK socket3511  6510 driver51e
SOCKET_NETWORK socket3512  6511 driver51f
SOCKET_NETWORK socket3513  6512 driver51a
SOCKET_NETWORK socket3514  6513 driver51b
SOCKET_NETWORK socket3515  6514 driver51c
SOCKET_NETWORK socket3516  6515 driver51d
SOCKET_NETWORK socket3517  6516 driver51e
SOCKET_NETWORK socket3518  6517 driver51f
SOCKET_NETWORK socket3519  6518 driver51a
SOCKET_NETWORK socket3520  6519 driver51b
```

SOCKET_NETWORK socket3521 6520 driver51c
SOCKET_NETWORK socket3522 6521 driver51d
SOCKET_NETWORK socket3523 6522 driver51e
SOCKET_NETWORK socket3524 6523 driver51f
SOCKET_NETWORK socket3525 6524 driver51a
SOCKET_NETWORK socket3526 6525 driver51b
SOCKET_NETWORK socket3527 6526 driver51c
SOCKET_NETWORK socket3528 6527 driver51d
SOCKET_NETWORK socket3529 6528 driver51e
SOCKET_NETWORK socket3530 6529 driver51f
SOCKET_NETWORK socket3531 6530 driver51a
SOCKET_NETWORK socket3532 6531 driver51b
SOCKET_NETWORK socket3533 6532 driver51c
SOCKET_NETWORK socket3534 6533 driver51d
SOCKET_NETWORK socket3535 6534 driver51e
SOCKET_NETWORK socket3536 6535 driver51f
SOCKET_NETWORK socket3537 6536 driver51a
SOCKET_NETWORK socket3538 6537 driver51b
SOCKET_NETWORK socket3539 6538 driver51c
SOCKET_NETWORK socket3540 6539 driver51d
SOCKET_NETWORK socket3541 6540 driver51e
SOCKET_NETWORK socket3542 6541 driver51f
SOCKET_NETWORK socket3543 6542 driver51a
SOCKET_NETWORK socket3544 6543 driver51b
SOCKET_NETWORK socket3545 6544 driver51c
SOCKET_NETWORK socket3546 6545 driver51d
SOCKET_NETWORK socket3547 6546 driver51e
SOCKET_NETWORK socket3548 6547 driver51f
SOCKET_NETWORK socket3549 6548 driver51a
SOCKET_NETWORK socket3550 6549 driver51b
SOCKET_NETWORK socket3551 6550 driver51c
SOCKET_NETWORK socket3552 6551 driver51d
SOCKET_NETWORK socket3553 6552 driver51e
SOCKET_NETWORK socket3554 6553 driver51f
SOCKET_NETWORK socket3555 6554 driver51a
SOCKET_NETWORK socket3556 6555 driver51b
SOCKET_NETWORK socket3557 6556 driver51c
SOCKET_NETWORK socket3558 6557 driver51d
SOCKET_NETWORK socket3559 6558 driver51e
SOCKET_NETWORK socket3560 6559 driver51f
SOCKET_NETWORK socket3561 6560 driver51a
SOCKET_NETWORK socket3562 6561 driver51b
SOCKET_NETWORK socket3563 6562 driver51c
SOCKET_NETWORK socket3564 6563 driver51d
SOCKET_NETWORK socket3565 6564 driver51e
SOCKET_NETWORK socket3566 6565 driver51f
SOCKET_NETWORK socket3567 6566 driver51a
SOCKET_NETWORK socket3568 6567 driver51b
SOCKET_NETWORK socket3569 6568 driver51c
SOCKET_NETWORK socket3570 6569 driver51d
SOCKET_NETWORK socket3571 6570 driver52a
SOCKET_NETWORK socket3572 6571 driver52b
SOCKET_NETWORK socket3573 6572 driver52c
SOCKET_NETWORK socket3574 6573 driver52d
SOCKET_NETWORK socket3575 6574 driver52e
SOCKET_NETWORK socket3576 6575 driver52f
SOCKET_NETWORK socket3577 6576 driver52a
SOCKET_NETWORK socket3578 6577 driver52b
SOCKET_NETWORK socket3579 6578 driver52c
SOCKET_NETWORK socket3580 6579 driver52d
SOCKET_NETWORK socket3581 6580 driver52e
SOCKET_NETWORK socket3582 6581 driver52f
SOCKET_NETWORK socket3583 6582 driver52a
SOCKET_NETWORK socket3584 6583 driver52b
SOCKET_NETWORK socket3585 6584 driver52c
SOCKET_NETWORK socket3586 6585 driver52d
SOCKET_NETWORK socket3587 6586 driver52e
SOCKET_NETWORK socket3588 6587 driver52f
SOCKET_NETWORK socket3589 6588 driver52a
SOCKET_NETWORK socket3590 6589 driver52b
SOCKET_NETWORK socket3591 6590 driver52c
SOCKET_NETWORK socket3592 6591 driver52d
SOCKET_NETWORK socket3593 6592 driver52e
SOCKET_NETWORK socket3594 6593 driver52f
SOCKET_NETWORK socket3595 6594 driver52a
SOCKET_NETWORK socket3596 6595 driver52b
SOCKET_NETWORK socket3597 6596 driver52c
SOCKET_NETWORK socket3598 6597 driver52d
SOCKET_NETWORK socket3599 6598 driver52e
SOCKET_NETWORK socket3600 6599 driver52f
SOCKET_NETWORK socket3601 6600 driver52a
SOCKET_NETWORK socket3602 6601 driver52b
SOCKET_NETWORK socket3603 6602 driver52c
SOCKET_NETWORK socket3604 6603 driver52d
SOCKET_NETWORK socket3605 6604 driver52e
SOCKET_NETWORK socket3606 6605 driver52f
SOCKET_NETWORK socket3607 6606 driver52a
SOCKET_NETWORK socket3608 6607 driver52b
SOCKET_NETWORK socket3609 6608 driver52c
SOCKET_NETWORK socket3610 6609 driver52d
SOCKET_NETWORK socket3611 6610 driver52e
SOCKET_NETWORK socket3612 6611 driver52f
SOCKET_NETWORK socket3613 6612 driver52a
SOCKET_NETWORK socket3614 6613 driver52b
SOCKET_NETWORK socket3615 6614 driver52c
SOCKET_NETWORK socket3616 6615 driver52d
SOCKET_NETWORK socket3617 6616 driver52e
SOCKET_NETWORK socket3618 6617 driver52f
SOCKET_NETWORK socket3619 6618 driver52a
SOCKET_NETWORK socket3620 6619 driver52b
SOCKET_NETWORK socket3621 6620 driver52c
SOCKET_NETWORK socket3622 6621 driver52d
SOCKET_NETWORK socket3623 6622 driver52e
SOCKET_NETWORK socket3624 6623 driver52f
SOCKET_NETWORK socket3625 6624 driver52a
SOCKET_NETWORK socket3626 6625 driver52b
SOCKET_NETWORK socket3627 6626 driver52c
SOCKET_NETWORK socket3628 6627 driver52d
SOCKET_NETWORK socket3629 6628 driver52e
SOCKET_NETWORK socket3630 6629 driver52f
SOCKET_NETWORK socket3631 6630 driver52a
SOCKET_NETWORK socket3632 6631 driver52b
SOCKET_NETWORK socket3633 6632 driver52c
SOCKET_NETWORK socket3634 6633 driver52d
SOCKET_NETWORK socket3635 6634 driver52e
SOCKET_NETWORK socket3636 6635 driver52f
SOCKET_NETWORK socket3637 6636 driver52a
SOCKET_NETWORK socket3638 6637 driver52b
SOCKET_NETWORK socket3639 6638 driver52c
SOCKET_NETWORK socket3640 6639 driver52d
SOCKET_NETWORK socket3641 6500 driver53a
SOCKET_NETWORK socket3642 6501 driver53b
SOCKET_NETWORK socket3643 6502 driver53c
SOCKET_NETWORK socket3644 6503 driver53d
SOCKET_NETWORK socket3645 6504 driver53e
SOCKET_NETWORK socket3646 6505 driver53f
SOCKET_NETWORK socket3647 6506 driver53a
SOCKET_NETWORK socket3648 6507 driver53b
SOCKET_NETWORK socket3649 6508 driver53c
SOCKET_NETWORK socket3650 6509 driver53d
SOCKET_NETWORK socket3651 6510 driver53e
SOCKET_NETWORK socket3652 6511 driver53f
SOCKET_NETWORK socket3653 6512 driver53a
SOCKET_NETWORK socket3654 6513 driver53b
SOCKET_NETWORK socket3655 6514 driver53c
SOCKET_NETWORK socket3656 6515 driver53d
SOCKET_NETWORK socket3657 6516 driver53e
SOCKET_NETWORK socket3658 6517 driver53f
SOCKET_NETWORK socket3659 6518 driver53a
SOCKET_NETWORK socket3660 6519 driver53b
SOCKET_NETWORK socket3661 6520 driver53c
SOCKET_NETWORK socket3662 6521 driver53d
SOCKET_NETWORK socket3663 6522 driver53e
SOCKET_NETWORK socket3664 6523 driver53f
SOCKET_NETWORK socket3665 6524 driver53a
SOCKET_NETWORK socket3666 6525 driver53b
SOCKET_NETWORK socket3667 6526 driver53c
SOCKET_NETWORK socket3668 6527 driver53d
SOCKET_NETWORK socket3669 6528 driver53e
SOCKET_NETWORK socket3670 6529 driver53f
SOCKET_NETWORK socket3671 6530 driver53a
SOCKET_NETWORK socket3672 6531 driver53b
SOCKET_NETWORK socket3673 6532 driver53c
SOCKET_NETWORK socket3674 6533 driver53d
SOCKET_NETWORK socket3675 6534 driver53e
SOCKET_NETWORK socket3676 6535 driver53f
SOCKET_NETWORK socket3677 6536 driver53a
SOCKET_NETWORK socket3678 6537 driver53b
SOCKET_NETWORK socket3679 6538 driver53c
SOCKET_NETWORK socket3680 6539 driver53d
SOCKET_NETWORK socket3681 6540 driver53e
SOCKET_NETWORK socket3682 6541 driver53f
SOCKET_NETWORK socket3683 6542 driver53a
SOCKET_NETWORK socket3684 6543 driver53b
SOCKET_NETWORK socket3685 6544 driver53c
SOCKET_NETWORK socket3686 6545 driver53d
SOCKET_NETWORK socket3687 6546 driver53e
SOCKET_NETWORK socket3688 6547 driver53f
SOCKET_NETWORK socket3689 6548 driver53a
SOCKET_NETWORK socket3690 6549 driver53b
SOCKET_NETWORK socket3691 6550 driver53c
SOCKET_NETWORK socket3692 6551 driver53d
SOCKET_NETWORK socket3693 6552 driver53e
SOCKET_NETWORK socket3694 6553 driver53f
SOCKET_NETWORK socket3695 6554 driver53a
SOCKET_NETWORK socket3696 6555 driver53b
SOCKET_NETWORK socket3697 6556 driver53c
SOCKET_NETWORK socket3698 6557 driver53d
SOCKET_NETWORK socket3699 6558 driver53e
SOCKET_NETWORK socket3700 6559 driver53f
SOCKET_NETWORK socket3701 6560 driver53a
SOCKET_NETWORK socket3702 6561 driver53b
SOCKET_NETWORK socket3703 6562 driver53c
SOCKET_NETWORK socket3704 6563 driver53d
SOCKET_NETWORK socket3705 6564 driver53e
SOCKET_NETWORK socket3706 6565 driver53f
SOCKET_NETWORK socket3707 6566 driver53a
SOCKET_NETWORK socket3708 6567 driver53b
SOCKET_NETWORK socket3709 6568 driver53c
SOCKET_NETWORK socket3710 6569 driver53d
SOCKET_NETWORK socket3711 6570 driver54a
SOCKET_NETWORK socket3712 6571 driver54b
SOCKET_NETWORK socket3713 6572 driver54c
SOCKET_NETWORK socket3714 6573 driver54d
SOCKET_NETWORK socket3715 6574 driver54e
SOCKET_NETWORK socket3716 6575 driver54f
SOCKET_NETWORK socket3717 6576 driver54a
SOCKET_NETWORK socket3718 6577 driver54b

```
SOCKET_NETWORK socket3719  6578 driver54c
SOCKET_NETWORK socket3720  6579 driver54d
SOCKET_NETWORK socket3721  6580 driver54e
SOCKET_NETWORK socket3722  6581 driver54f
SOCKET_NETWORK socket3723  6582 driver54a
SOCKET_NETWORK socket3724  6583 driver54b
SOCKET_NETWORK socket3725  6584 driver54c
SOCKET_NETWORK socket3726  6585 driver54d
SOCKET_NETWORK socket3727  6586 driver54e
SOCKET_NETWORK socket3728  6587 driver54f
SOCKET_NETWORK socket3729  6588 driver54a
SOCKET_NETWORK socket3730  6589 driver54b
SOCKET_NETWORK socket3731  6590 driver54c
SOCKET_NETWORK socket3732  6591 driver54d
SOCKET_NETWORK socket3733  6592 driver54e
SOCKET_NETWORK socket3734  6593 driver54f
SOCKET_NETWORK socket3735  6594 driver54a
SOCKET_NETWORK socket3736  6595 driver54b
SOCKET_NETWORK socket3737  6596 driver54c
SOCKET_NETWORK socket3738  6597 driver54d
SOCKET_NETWORK socket3739  6598 driver54e
SOCKET_NETWORK socket3740  6599 driver54f
SOCKET_NETWORK socket3741  6600 driver54a
SOCKET_NETWORK socket3742  6601 driver54b
SOCKET_NETWORK socket3743  6602 driver54c
SOCKET_NETWORK socket3744  6603 driver54d
SOCKET_NETWORK socket3745  6604 driver54e
SOCKET_NETWORK socket3746  6605 driver54f
SOCKET_NETWORK socket3747  6606 driver54a
SOCKET_NETWORK socket3748  6607 driver54b
SOCKET_NETWORK socket3749  6608 driver54c
SOCKET_NETWORK socket3750  6609 driver54d
SOCKET_NETWORK socket3751  6610 driver54e
SOCKET_NETWORK socket3752  6611 driver54f
SOCKET_NETWORK socket3753  6612 driver54a
SOCKET_NETWORK socket3754  6613 driver54b
SOCKET_NETWORK socket3755  6614 driver54c
SOCKET_NETWORK socket3756  6615 driver54d
SOCKET_NETWORK socket3757  6616 driver54e
SOCKET_NETWORK socket3758  6617 driver54f
SOCKET_NETWORK socket3759  6618 driver54a
SOCKET_NETWORK socket3760  6619 driver54b
SOCKET_NETWORK socket3761  6620 driver54c
SOCKET_NETWORK socket3762  6621 driver54d
SOCKET_NETWORK socket3763  6622 driver54e
SOCKET_NETWORK socket3764  6623 driver54f
SOCKET_NETWORK socket3765  6624 driver54a
SOCKET_NETWORK socket3766  6625 driver54b
SOCKET_NETWORK socket3767  6626 driver54c
SOCKET_NETWORK socket3768  6627 driver54d
SOCKET_NETWORK socket3769  6628 driver54e
SOCKET_NETWORK socket3770  6629 driver54f
SOCKET_NETWORK socket3771  6630 driver54a
SOCKET_NETWORK socket3772  6631 driver54b
SOCKET_NETWORK socket3773  6632 driver54c
SOCKET_NETWORK socket3774  6633 driver54d
SOCKET_NETWORK socket3775  6634 driver54e
SOCKET_NETWORK socket3776  6635 driver54f
SOCKET_NETWORK socket3777  6636 driver54a
SOCKET_NETWORK socket3778  6637 driver54b
SOCKET_NETWORK socket3779  6638 driver54c
SOCKET_NETWORK socket3780  6639 driver54d
#elif MASTER_NUM10
SOCKET_NETWORK socket3781  6500 driver55a
SOCKET_NETWORK socket3782  6501 driver55b
SOCKET_NETWORK socket3783  6502 driver55c
SOCKET_NETWORK socket3784  6503 driver55d
SOCKET_NETWORK socket3785  6504 driver55e
SOCKET_NETWORK socket3786  6505 driver55f
SOCKET_NETWORK socket3787  6506 driver55a
SOCKET_NETWORK socket3788  6507 driver55b
SOCKET_NETWORK socket3789  6508 driver55c
SOCKET_NETWORK socket3790  6509 driver55d
SOCKET_NETWORK socket3791  6510 driver55e
SOCKET_NETWORK socket3792  6511 driver55f
SOCKET_NETWORK socket3793  6512 driver55a
SOCKET_NETWORK socket3794  6513 driver55b
SOCKET_NETWORK socket3795  6514 driver55c
SOCKET_NETWORK socket3796  6515 driver55d
SOCKET_NETWORK socket3797  6516 driver55e
SOCKET_NETWORK socket3798  6517 driver55f
SOCKET_NETWORK socket3799  6518 driver55a
SOCKET_NETWORK socket3800  6519 driver55b
SOCKET_NETWORK socket3801  6520 driver55c
SOCKET_NETWORK socket3802  6521 driver55d
SOCKET_NETWORK socket3803  6522 driver55e
SOCKET_NETWORK socket3804  6523 driver55f
SOCKET_NETWORK socket3805  6524 driver55a
SOCKET_NETWORK socket3806  6525 driver55b
SOCKET_NETWORK socket3807  6526 driver55c
SOCKET_NETWORK socket3808  6527 driver55d
SOCKET_NETWORK socket3809  6528 driver55e
SOCKET_NETWORK socket3810  6529 driver55f
SOCKET_NETWORK socket3811  6530 driver55a
SOCKET_NETWORK socket3812  6531 driver55b
SOCKET_NETWORK socket3813  6532 driver55c
SOCKET_NETWORK socket3814  6533 driver55d
SOCKET_NETWORK socket3815  6534 driver55e
SOCKET_NETWORK socket3816  6535 driver55f
SOCKET_NETWORK socket3817  6536 driver55a
SOCKET_NETWORK socket3818  6537 driver55b
SOCKET_NETWORK socket3819  6538 driver55c
SOCKET_NETWORK socket3820  6539 driver55d
SOCKET_NETWORK socket3821  6540 driver55e
SOCKET_NETWORK socket3822  6541 driver55f
SOCKET_NETWORK socket3823  6542 driver55a
SOCKET_NETWORK socket3824  6543 driver55b
SOCKET_NETWORK socket3825  6544 driver55c
SOCKET_NETWORK socket3826  6545 driver55d
SOCKET_NETWORK socket3827  6546 driver55e
SOCKET_NETWORK socket3828  6547 driver55f
SOCKET_NETWORK socket3829  6548 driver55a
SOCKET_NETWORK socket3830  6549 driver55b
SOCKET_NETWORK socket3831  6550 driver55c
SOCKET_NETWORK socket3832  6551 driver55d
SOCKET_NETWORK socket3833  6552 driver55e
SOCKET_NETWORK socket3834  6553 driver55f
SOCKET_NETWORK socket3835  6554 driver55a
SOCKET_NETWORK socket3836  6555 driver55b
SOCKET_NETWORK socket3837  6556 driver55c
SOCKET_NETWORK socket3838  6557 driver55d
SOCKET_NETWORK socket3839  6558 driver55e
SOCKET_NETWORK socket3840  6559 driver55f
SOCKET_NETWORK socket3841  6560 driver55a
SOCKET_NETWORK socket3842  6561 driver55b
SOCKET_NETWORK socket3843  6562 driver55c
SOCKET_NETWORK socket3844  6563 driver55d
SOCKET_NETWORK socket3845  6564 driver55e
SOCKET_NETWORK socket3846  6565 driver55f
SOCKET_NETWORK socket3847  6566 driver55a
SOCKET_NETWORK socket3848  6567 driver55b
SOCKET_NETWORK socket3849  6568 driver55c
SOCKET_NETWORK socket3850  6569 driver55d
SOCKET_NETWORK socket3851  6570 driver56a
SOCKET_NETWORK socket3852  6571 driver56b
SOCKET_NETWORK socket3853  6572 driver56c
SOCKET_NETWORK socket3854  6573 driver56d
SOCKET_NETWORK socket3855  6574 driver56e
SOCKET_NETWORK socket3856  6575 driver56f
SOCKET_NETWORK socket3857  6576 driver56a
SOCKET_NETWORK socket3858  6577 driver56b
SOCKET_NETWORK socket3859  6578 driver56c
SOCKET_NETWORK socket3860  6579 driver56d
SOCKET_NETWORK socket3861  6580 driver56e
SOCKET_NETWORK socket3862  6581 driver56f
SOCKET_NETWORK socket3863  6582 driver56a
SOCKET_NETWORK socket3864  6583 driver56b
SOCKET_NETWORK socket3865  6584 driver56c
SOCKET_NETWORK socket3866  6585 driver56d
SOCKET_NETWORK socket3867  6586 driver56e
SOCKET_NETWORK socket3868  6587 driver56f
SOCKET_NETWORK socket3869  6588 driver56a
SOCKET_NETWORK socket3870  6589 driver56b
SOCKET_NETWORK socket3871  6590 driver56c
SOCKET_NETWORK socket3872  6591 driver56d
SOCKET_NETWORK socket3873  6592 driver56e
SOCKET_NETWORK socket3874  6593 driver56f
SOCKET_NETWORK socket3875  6594 driver56a
SOCKET_NETWORK socket3876  6595 driver56b
SOCKET_NETWORK socket3877  6596 driver56c
SOCKET_NETWORK socket3878  6597 driver56d
SOCKET_NETWORK socket3879  6598 driver56e
SOCKET_NETWORK socket3880  6599 driver56f
SOCKET_NETWORK socket3881  6600 driver56a
SOCKET_NETWORK socket3882  6601 driver56b
SOCKET_NETWORK socket3883  6602 driver56c
SOCKET_NETWORK socket3884  6603 driver56d
SOCKET_NETWORK socket3885  6604 driver56e
SOCKET_NETWORK socket3886  6605 driver56f
SOCKET_NETWORK socket3887  6606 driver56a
SOCKET_NETWORK socket3888  6607 driver56b
SOCKET_NETWORK socket3889  6608 driver56c
SOCKET_NETWORK socket3890  6609 driver56d
SOCKET_NETWORK socket3891  6610 driver56e
SOCKET_NETWORK socket3892  6611 driver56f
SOCKET_NETWORK socket3893  6612 driver56a
SOCKET_NETWORK socket3894  6613 driver56b
SOCKET_NETWORK socket3895  6614 driver56c
SOCKET_NETWORK socket3896  6615 driver56d
SOCKET_NETWORK socket3897  6616 driver56e
SOCKET_NETWORK socket3898  6617 driver56f
SOCKET_NETWORK socket3899  6618 driver56a
SOCKET_NETWORK socket3900  6619 driver56b
SOCKET_NETWORK socket3901  6620 driver56c
SOCKET_NETWORK socket3902  6621 driver56d
SOCKET_NETWORK socket3903  6622 driver56e
SOCKET_NETWORK socket3904  6623 driver56f
SOCKET_NETWORK socket3905  6624 driver56a
SOCKET_NETWORK socket3906  6625 driver56b
SOCKET_NETWORK socket3907  6626 driver56c
SOCKET_NETWORK socket3908  6627 driver56d
SOCKET_NETWORK socket3909  6628 driver56e
SOCKET_NETWORK socket3910  6629 driver56f
SOCKET_NETWORK socket3911  6630 driver56a
SOCKET_NETWORK socket3912  6631 driver56b
SOCKET_NETWORK socket3913  6632 driver56c
SOCKET_NETWORK socket3914  6633 driver56d
SOCKET_NETWORK socket3915  6634 driver56e
```

SOCKET_NETWORK socket3916 6635 driver56f
SOCKET_NETWORK socket3917 6636 driver56a
SOCKET_NETWORK socket3918 6637 driver56b
SOCKET_NETWORK socket3919 6638 driver56c
SOCKET_NETWORK socket3920 6639 driver56d
SOCKET_NETWORK socket3921 6500 driver57a
SOCKET_NETWORK socket3922 6501 driver57b
SOCKET_NETWORK socket3923 6502 driver57c
SOCKET_NETWORK socket3924 6503 driver57d
SOCKET_NETWORK socket3925 6504 driver57e
SOCKET_NETWORK socket3926 6505 driver57f
SOCKET_NETWORK socket3927 6506 driver57a
SOCKET_NETWORK socket3928 6507 driver57b
SOCKET_NETWORK socket3929 6508 driver57c
SOCKET_NETWORK socket3930 6509 driver57d
SOCKET_NETWORK socket3931 6510 driver57e
SOCKET_NETWORK socket3932 6511 driver57f
SOCKET_NETWORK socket3933 6512 driver57a
SOCKET_NETWORK socket3934 6513 driver57b
SOCKET_NETWORK socket3935 6514 driver57c
SOCKET_NETWORK socket3936 6515 driver57d
SOCKET_NETWORK socket3937 6516 driver57e
SOCKET_NETWORK socket3938 6517 driver57f
SOCKET_NETWORK socket3939 6518 driver57a
SOCKET_NETWORK socket3940 6519 driver57b
SOCKET_NETWORK socket3941 6520 driver57c
SOCKET_NETWORK socket3942 6521 driver57d
SOCKET_NETWORK socket3943 6522 driver57e
SOCKET_NETWORK socket3944 6523 driver57f
SOCKET_NETWORK socket3945 6524 driver57a
SOCKET_NETWORK socket3946 6525 driver57b
SOCKET_NETWORK socket3947 6526 driver57c
SOCKET_NETWORK socket3948 6527 driver57d
SOCKET_NETWORK socket3949 6528 driver57e
SOCKET_NETWORK socket3950 6529 driver57f
SOCKET_NETWORK socket3951 6530 driver57a
SOCKET_NETWORK socket3952 6531 driver57b
SOCKET_NETWORK socket3953 6532 driver57c
SOCKET_NETWORK socket3954 6533 driver57d
SOCKET_NETWORK socket3955 6534 driver57e
SOCKET_NETWORK socket3956 6535 driver57f
SOCKET_NETWORK socket3957 6536 driver57a
SOCKET_NETWORK socket3958 6537 driver57b
SOCKET_NETWORK socket3959 6538 driver57c
SOCKET_NETWORK socket3960 6539 driver57d
SOCKET_NETWORK socket3961 6540 driver57e
SOCKET_NETWORK socket3962 6541 driver57f
SOCKET_NETWORK socket3963 6542 driver57a
SOCKET_NETWORK socket3964 6543 driver57b
SOCKET_NETWORK socket3965 6544 driver57c
SOCKET_NETWORK socket3966 6545 driver57d
SOCKET_NETWORK socket3967 6546 driver57e
SOCKET_NETWORK socket3968 6547 driver57f
SOCKET_NETWORK socket3969 6548 driver57a
SOCKET_NETWORK socket3970 6549 driver57b
SOCKET_NETWORK socket3971 6550 driver57c
SOCKET_NETWORK socket3972 6551 driver57d
SOCKET_NETWORK socket3973 6552 driver57e
SOCKET_NETWORK socket3974 6553 driver57f
SOCKET_NETWORK socket3975 6554 driver57a
SOCKET_NETWORK socket3976 6555 driver57b
SOCKET_NETWORK socket3977 6556 driver57c
SOCKET_NETWORK socket3978 6557 driver57d
SOCKET_NETWORK socket3979 6558 driver57e
SOCKET_NETWORK socket3980 6559 driver57f
SOCKET_NETWORK socket3981 6560 driver57a
SOCKET_NETWORK socket3982 6561 driver57b
SOCKET_NETWORK socket3983 6562 driver57c
SOCKET_NETWORK socket3984 6563 driver57d
SOCKET_NETWORK socket3985 6564 driver57e
SOCKET_NETWORK socket3986 6565 driver57f
SOCKET_NETWORK socket3987 6566 driver57a
SOCKET_NETWORK socket3988 6567 driver57b
SOCKET_NETWORK socket3989 6568 driver57c
SOCKET_NETWORK socket3990 6569 driver57d
SOCKET_NETWORK socket3991 6570 driver58a
SOCKET_NETWORK socket3992 6571 driver58b
SOCKET_NETWORK socket3993 6572 driver58c
SOCKET_NETWORK socket3994 6573 driver58d
SOCKET_NETWORK socket3995 6574 driver58e
SOCKET_NETWORK socket3996 6575 driver58f
SOCKET_NETWORK socket3997 6576 driver58a
SOCKET_NETWORK socket3998 6577 driver58b
SOCKET_NETWORK socket3999 6578 driver58c
SOCKET_NETWORK socket4000 6579 driver58d
SOCKET_NETWORK socket4001 6580 driver58e
SOCKET_NETWORK socket4002 6581 driver58f
SOCKET_NETWORK socket4003 6582 driver58a
SOCKET_NETWORK socket4004 6583 driver58b
SOCKET_NETWORK socket4005 6584 driver58c
SOCKET_NETWORK socket4006 6585 driver58d
SOCKET_NETWORK socket4007 6586 driver58e
SOCKET_NETWORK socket4008 6587 driver58f
SOCKET_NETWORK socket4009 6588 driver58a
SOCKET_NETWORK socket4010 6589 driver58b
SOCKET_NETWORK socket4011 6590 driver58c
SOCKET_NETWORK socket4012 6591 driver58d
SOCKET_NETWORK socket4013 6592 driver58e
SOCKET_NETWORK socket4014 6593 driver58f
SOCKET_NETWORK socket4015 6594 driver58a
SOCKET_NETWORK socket4016 6595 driver58b
SOCKET_NETWORK socket4017 6596 driver58c
SOCKET_NETWORK socket4018 6597 driver58d
SOCKET_NETWORK socket4019 6598 driver58e
SOCKET_NETWORK socket4020 6599 driver58f
SOCKET_NETWORK socket4021 6600 driver58a
SOCKET_NETWORK socket4022 6601 driver58b
SOCKET_NETWORK socket4023 6602 driver58c
SOCKET_NETWORK socket4024 6603 driver58d
SOCKET_NETWORK socket4025 6604 driver58e
SOCKET_NETWORK socket4026 6605 driver58f
SOCKET_NETWORK socket4027 6606 driver58a
SOCKET_NETWORK socket4028 6607 driver58b
SOCKET_NETWORK socket4029 6608 driver58c
SOCKET_NETWORK socket4030 6609 driver58d
SOCKET_NETWORK socket4031 6610 driver58e
SOCKET_NETWORK socket4032 6611 driver58f
SOCKET_NETWORK socket4033 6612 driver58a
SOCKET_NETWORK socket4034 6613 driver58b
SOCKET_NETWORK socket4035 6614 driver58c
SOCKET_NETWORK socket4036 6615 driver58d
SOCKET_NETWORK socket4037 6616 driver58e
SOCKET_NETWORK socket4038 6617 driver58f
SOCKET_NETWORK socket4039 6618 driver58a
SOCKET_NETWORK socket4040 6619 driver58b
SOCKET_NETWORK socket4041 6620 driver58c
SOCKET_NETWORK socket4042 6621 driver58d
SOCKET_NETWORK socket4043 6622 driver58e
SOCKET_NETWORK socket4044 6623 driver58f
SOCKET_NETWORK socket4045 6624 driver58a
SOCKET_NETWORK socket4046 6625 driver58b
SOCKET_NETWORK socket4047 6626 driver58c
SOCKET_NETWORK socket4048 6627 driver58d
SOCKET_NETWORK socket4049 6628 driver58e
SOCKET_NETWORK socket4050 6629 driver58f
SOCKET_NETWORK socket4051 6630 driver58a
SOCKET_NETWORK socket4052 6631 driver58b
SOCKET_NETWORK socket4053 6632 driver58c
SOCKET_NETWORK socket4054 6633 driver58d
SOCKET_NETWORK socket4055 6634 driver58e
SOCKET_NETWORK socket4056 6635 driver58f
SOCKET_NETWORK socket4057 6636 driver58a
SOCKET_NETWORK socket4058 6637 driver58b
SOCKET_NETWORK socket4059 6638 driver58c
SOCKET_NETWORK socket4060 6639 driver58d
SOCKET_NETWORK socket4061 6500 driver59a
SOCKET_NETWORK socket4062 6501 driver59b
SOCKET_NETWORK socket4063 6502 driver59c
SOCKET_NETWORK socket4064 6503 driver59d
SOCKET_NETWORK socket4065 6504 driver59e
SOCKET_NETWORK socket4066 6505 driver59f
SOCKET_NETWORK socket4067 6506 driver59a
SOCKET_NETWORK socket4068 6507 driver59b
SOCKET_NETWORK socket4069 6508 driver59c
SOCKET_NETWORK socket4070 6509 driver59d
SOCKET_NETWORK socket4071 6510 driver59e
SOCKET_NETWORK socket4072 6511 driver59f
SOCKET_NETWORK socket4073 6512 driver59a
SOCKET_NETWORK socket4074 6513 driver59b
SOCKET_NETWORK socket4075 6514 driver59c
SOCKET_NETWORK socket4076 6515 driver59d
SOCKET_NETWORK socket4077 6516 driver59e
SOCKET_NETWORK socket4078 6517 driver59f
SOCKET_NETWORK socket4079 6518 driver59a
SOCKET_NETWORK socket4080 6519 driver59b
SOCKET_NETWORK socket4081 6520 driver59c
SOCKET_NETWORK socket4082 6521 driver59d
SOCKET_NETWORK socket4083 6522 driver59e
SOCKET_NETWORK socket4084 6523 driver59f
SOCKET_NETWORK socket4085 6524 driver59a
SOCKET_NETWORK socket4086 6525 driver59b
SOCKET_NETWORK socket4087 6526 driver59c
SOCKET_NETWORK socket4088 6527 driver59d
SOCKET_NETWORK socket4089 6528 driver59e
SOCKET_NETWORK socket4090 6529 driver59f
SOCKET_NETWORK socket4091 6530 driver59a
SOCKET_NETWORK socket4092 6531 driver59b
SOCKET_NETWORK socket4093 6532 driver59c
SOCKET_NETWORK socket4094 6533 driver59d
SOCKET_NETWORK socket4095 6534 driver59e
SOCKET_NETWORK socket4096 6535 driver59f
SOCKET_NETWORK socket4097 6536 driver59a
SOCKET_NETWORK socket4098 6537 driver59b
SOCKET_NETWORK socket4099 6538 driver59c
SOCKET_NETWORK socket4100 6539 driver59d
SOCKET_NETWORK socket4101 6540 driver59e
SOCKET_NETWORK socket4102 6541 driver59f
SOCKET_NETWORK socket4103 6542 driver59a
SOCKET_NETWORK socket4104 6543 driver59b
SOCKET_NETWORK socket4105 6544 driver59c
SOCKET_NETWORK socket4106 6545 driver59d
SOCKET_NETWORK socket4107 6546 driver59e
SOCKET_NETWORK socket4108 6547 driver59f
SOCKET_NETWORK socket4109 6548 driver59a
SOCKET_NETWORK socket4110 6549 driver59b
SOCKET_NETWORK socket4111 6550 driver59c
SOCKET_NETWORK socket4112 6551 driver59d
SOCKET_NETWORK socket4113 6552 driver59e

```
SOCKET_NETWORK socket4114   6553 driver59f
SOCKET_NETWORK socket4115   6554 driver59a
SOCKET_NETWORK socket4116   6555 driver59b
SOCKET_NETWORK socket4117   6556 driver59c
SOCKET_NETWORK socket4118   6557 driver59d
SOCKET_NETWORK socket4119   6558 driver59e
SOCKET_NETWORK socket4120   6559 driver59f
SOCKET_NETWORK socket4121   6560 driver59a
SOCKET_NETWORK socket4122   6561 driver59b
SOCKET_NETWORK socket4123   6562 driver59c
SOCKET_NETWORK socket4124   6563 driver59d
SOCKET_NETWORK socket4125   6564 driver59e
SOCKET_NETWORK socket4126   6565 driver59f
SOCKET_NETWORK socket4127   6566 driver59a
SOCKET_NETWORK socket4128   6567 driver59b
SOCKET_NETWORK socket4129   6568 driver59c
SOCKET_NETWORK socket4130   6569 driver59d
SOCKET_NETWORK socket4131   6570 driver60a
SOCKET_NETWORK socket4132   6571 driver60b
SOCKET_NETWORK socket4133   6572 driver60c
SOCKET_NETWORK socket4134   6573 driver60d
SOCKET_NETWORK socket4135   6574 driver60e
SOCKET_NETWORK socket4136   6575 driver60f
SOCKET_NETWORK socket4137   6576 driver60a
SOCKET_NETWORK socket4138   6577 driver60b
SOCKET_NETWORK socket4139   6578 driver60c
SOCKET_NETWORK socket4140   6579 driver60d
SOCKET_NETWORK socket4141   6580 driver60e
SOCKET_NETWORK socket4142   6581 driver60f
SOCKET_NETWORK socket4143   6582 driver60a
SOCKET_NETWORK socket4144   6583 driver60b
SOCKET_NETWORK socket4145   6584 driver60c
SOCKET_NETWORK socket4146   6585 driver60d
SOCKET_NETWORK socket4147   6586 driver60e
SOCKET_NETWORK socket4148   6587 driver60f
SOCKET_NETWORK socket4149   6588 driver60a
SOCKET_NETWORK socket4150   6589 driver60b
SOCKET_NETWORK socket4151   6590 driver60c
SOCKET_NETWORK socket4152   6591 driver60d
SOCKET_NETWORK socket4153   6592 driver60e
SOCKET_NETWORK socket4154   6593 driver60f
SOCKET_NETWORK socket4155   6594 driver60a
SOCKET_NETWORK socket4156   6595 driver60b
SOCKET_NETWORK socket4157   6596 driver60c
SOCKET_NETWORK socket4158   6597 driver60d
SOCKET_NETWORK socket4159   6598 driver60e
SOCKET_NETWORK socket4160   6599 driver60f
SOCKET_NETWORK socket4161   6600 driver60a
SOCKET_NETWORK socket4162   6601 driver60b
SOCKET_NETWORK socket4163   6602 driver60c
SOCKET_NETWORK socket4164   6603 driver60d
SOCKET_NETWORK socket4165   6604 driver60e
SOCKET_NETWORK socket4166   6605 driver60f
SOCKET_NETWORK socket4167   6606 driver60a
SOCKET_NETWORK socket4168   6607 driver60b
SOCKET_NETWORK socket4169   6608 driver60c
SOCKET_NETWORK socket4170   6609 driver60d
SOCKET_NETWORK socket4171   6610 driver60e
SOCKET_NETWORK socket4172   6611 driver60f
SOCKET_NETWORK socket4173   6612 driver60a
SOCKET_NETWORK socket4174   6613 driver60b
SOCKET_NETWORK socket4175   6614 driver60c
SOCKET_NETWORK socket4176   6615 driver60d
SOCKET_NETWORK socket4177   6616 driver60e
SOCKET_NETWORK socket4178   6617 driver60f
SOCKET_NETWORK socket4179   6618 driver60a
SOCKET_NETWORK socket4180   6619 driver60b
SOCKET_NETWORK socket4181   6620 driver60c
SOCKET_NETWORK socket4182   6621 driver60d
SOCKET_NETWORK socket4183   6622 driver60e
SOCKET_NETWORK socket4184   6623 driver60f
SOCKET_NETWORK socket4185   6624 driver60a
SOCKET_NETWORK socket4186   6625 driver60b
SOCKET_NETWORK socket4187   6626 driver60c
SOCKET_NETWORK socket4188   6627 driver60d
SOCKET_NETWORK socket4189   6628 driver60e
SOCKET_NETWORK socket4190   6629 driver60f
SOCKET_NETWORK socket4191   6630 driver60a
SOCKET_NETWORK socket4192   6631 driver60b
SOCKET_NETWORK socket4193   6632 driver60c
SOCKET_NETWORK socket4194   6633 driver60d
SOCKET_NETWORK socket4195   6634 driver60e
SOCKET_NETWORK socket4196   6635 driver60f
SOCKET_NETWORK socket4197   6636 driver60a
SOCKET_NETWORK socket4198   6637 driver60b
SOCKET_NETWORK socket4199   6638 driver60c
SOCKET_NETWORK socket4200   6639 driver60d
#endif
/*-----------------------------------------------------------------*/
OUTPUTNAME="regattaH"
CPU=48
BEGIN_WAIT=20:00
RAMPUP=30:00
RUNTIME=130:00
RAMPDOWN_WAIT=5:00
RAMPDOWN=17:00
/*NOTE ** MUST SET CONVERSIONS AS WELL */
/*RAMPUP_SEC    1800 */
/*RUNTIME_SEC   7800 */
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 140 0-14
START_RANGE client1b socket2 150 14-29
START_RANGE client1c socket3 140 29-43
START_RANGE client1d socket4 150 43-58
START_RANGE client1e socket5 140 58-72
START_RANGE client1f socket6 150 72-87
START_RANGE client1g socket7 140 87-101
START_RANGE client1h socket8 150 101-116
START_RANGE client1i socket9 140 116-130
START_RANGE client1j socket10 150 130-145
START_RANGE client1k socket11 140 145-159
START_RANGE client1l socket12 150 159-174
START_RANGE client1m socket13 140 174-188
START_RANGE client1n socket14 150 188-203
START_RANGE client1a socket15 140 203-217
START_RANGE client1b socket16 150 217-232
START_RANGE client1c socket17 140 232-246
START_RANGE client1d socket18 150 246-261
START_RANGE client1e socket19 140 261-275
START_RANGE client1f socket20 150 275-290
START_RANGE client1g socket21 140 290-304
START_RANGE client1h socket22 150 304-319
START_RANGE client1i socket23 140 319-333
START_RANGE client1j socket24 150 333-348
START_RANGE client1k socket25 140 348-362
START_RANGE client1l socket26 150 362-377
START_RANGE client1m socket27 140 377-391
START_RANGE client1n socket28 150 391-406
START_RANGE client1a socket29 140 406-420
START_RANGE client1b socket30 150 420-435
START_RANGE client1c socket31 140 435-449
START_RANGE client1d socket32 150 449-464
START_RANGE client1e socket33 140 464-478
START_RANGE client1f socket34 150 478-493
START_RANGE client1g socket35 140 493-507
START_RANGE client1h socket36 150 507-522
START_RANGE client1i socket37 140 522-536
START_RANGE client1j socket38 150 536-551
START_RANGE client1k socket39 140 551-565
START_RANGE client1l socket40 150 565-580
START_RANGE client1m socket41 140 580-594
START_RANGE client1n socket42 150 594-609
START_RANGE client1a socket43 140 609-623
START_RANGE client1b socket44 150 623-638
START_RANGE client1c socket45 140 638-652
START_RANGE client1d socket46 150 652-667
START_RANGE client1e socket47 140 667-681
START_RANGE client1f socket48 150 681-696
START_RANGE client1g socket49 140 696-710
START_RANGE client1h socket50 150 710-725
START_RANGE client1i socket51 140 725-739
START_RANGE client1j socket52 150 739-754
START_RANGE client1k socket53 140 754-768
START_RANGE client1l socket54 150 768-783
START_RANGE client1m socket55 140 783-797
START_RANGE client1n socket56 150 797-812
START_RANGE client1a socket57 140 812-826
START_RANGE client1b socket58 150 826-841
START_RANGE client1c socket59 140 841-855
START_RANGE client1d socket60 150 855-870
START_RANGE client1e socket61 140 870-884
START_RANGE client1f socket62 150 884-899
START_RANGE client1g socket63 140 899-913
START_RANGE client1h socket64 150 913-928
START_RANGE client1i socket65 140 928-942
START_RANGE client1j socket66 150 942-957
START_RANGE client1k socket67 140 957-971
START_RANGE client1l socket68 150 971-986
```

```
START_RANGE client1m socket69 140 986-1000
START_RANGE client1n socket70 150 1000-1015

START_RANGE client1o socket71 140 1015-1029
START_RANGE client1p socket72 150 1029-1044
START_RANGE client1q socket73 140 1044-1058
START_RANGE client1r socket74 150 1058-1073
START_RANGE client1s socket75 140 1073-1087
START_RANGE client1t socket76 150 1087-1102
START_RANGE client1u socket77 140 1102-1116
START_RANGE client1v socket78 150 1116-1131
START_RANGE client1w socket79 140 1131-1145
START_RANGE client1x socket80 150 1145-1160
START_RANGE client1y socket81 140 1160-1174
START_RANGE client1z socket82 150 1174-1189
START_RANGE client1aa socket83 140 1189-1203
START_RANGE client1ab socket84 150 1203-1218
START_RANGE client1o socket85 140 1218-1232
START_RANGE client1p socket86 150 1232-1247
START_RANGE client1q socket87 140 1247-1261
START_RANGE client1r socket88 150 1261-1276
START_RANGE client1s socket89 140 1276-1290
START_RANGE client1t socket90 150 1290-1305
START_RANGE client1u socket91 140 1305-1319
START_RANGE client1v socket92 150 1319-1334
START_RANGE client1w socket93 140 1334-1348
START_RANGE client1x socket94 150 1348-1363
START_RANGE client1y socket95 140 1363-1377
START_RANGE client1z socket96 150 1377-1392
START_RANGE client1aa socket97 140 1392-1406
START_RANGE client1ab socket98 150 1406-1421
START_RANGE client1o socket99 140 1421-1435
START_RANGE client1p socket100 150 1435-1450
START_RANGE client1q socket101 140 1450-1464
START_RANGE client1r socket102 150 1464-1479
START_RANGE client1s socket103 140 1479-1493
START_RANGE client1t socket104 150 1493-1508
START_RANGE client1u socket105 140 1508-1522
START_RANGE client1v socket106 150 1522-1537
START_RANGE client1w socket107 140 1537-1551
START_RANGE client1x socket108 150 1551-1566
START_RANGE client1y socket109 140 1566-1580
START_RANGE client1z socket110 150 1580-1595
START_RANGE client1aa socket111 140 1595-1609
START_RANGE client1ab socket112 150 1609-1624
START_RANGE client1o socket113 140 1624-1638
START_RANGE client1p socket114 150 1638-1653
START_RANGE client1q socket115 140 1653-1667
START_RANGE client1r socket116 150 1667-1682
START_RANGE client1s socket117 140 1682-1696
START_RANGE client1t socket118 150 1696-1711
START_RANGE client1u socket119 140 1711-1725
START_RANGE client1v socket120 150 1725-1740
START_RANGE client1w socket121 140 1740-1754
START_RANGE client1x socket122 150 1754-1769
START_RANGE client1y socket123 140 1769-1783
START_RANGE client1z socket124 150 1783-1798
START_RANGE client1aa socket125 140 1798-1812
START_RANGE client1ab socket126 150 1812-1827
START_RANGE client1o socket127 140 1827-1841
START_RANGE client1p socket128 150 1841-1856
START_RANGE client1q socket129 140 1856-1870
START_RANGE client1r socket130 150 1870-1885
START_RANGE client1s socket131 140 1885-1899
START_RANGE client1t socket132 150 1899-1914
START_RANGE client1u socket133 140 1914-1928
START_RANGE client1v socket134 150 1928-1943
START_RANGE client1w socket135 140 1943-1957
START_RANGE client1x socket136 150 1957-1972
START_RANGE client1y socket137 140 1972-1986
START_RANGE client1z socket138 150 1986-2001
START_RANGE client1aa socket139 140 2001-2015
START_RANGE client1ab socket140 150 2015-2030

START_RANGE client2a socket141 140 2030-2044
START_RANGE client2b socket142 150 2044-2059
START_RANGE client2c socket143 140 2059-2073
START_RANGE client2d socket144 150 2073-2088
START_RANGE client2e socket145 140 2088-2102
START_RANGE client2f socket146 150 2102-2117
START_RANGE client2g socket147 140 2117-2131
START_RANGE client2h socket148 150 2131-2146
START_RANGE client2i socket149 140 2146-2160
START_RANGE client2j socket150 150 2160-2175
START_RANGE client2k socket151 140 2175-2189
START_RANGE client2l socket152 150 2189-2204
START_RANGE client2m socket153 140 2204-2218
START_RANGE client2n socket154 150 2218-2233
START_RANGE client2a socket155 140 2233-2247
START_RANGE client2b socket156 150 2247-2262
START_RANGE client2c socket157 140 2262-2276
START_RANGE client2d socket158 150 2276-2291
START_RANGE client2e socket159 140 2291-2305
START_RANGE client2f socket160 150 2305-2320
START_RANGE client2g socket161 140 2320-2334
START_RANGE client2h socket162 150 2334-2349
START_RANGE client2i socket163 140 2349-2363
START_RANGE client2j socket164 150 2363-2378
START_RANGE client2k socket165 140 2378-2392
START_RANGE client2l socket166 150 2392-2407
START_RANGE client2m socket167 140 2407-2421
START_RANGE client2n socket168 150 2421-2436
START_RANGE client2a socket169 140 2436-2450
START_RANGE client2b socket170 150 2450-2465
START_RANGE client2c socket171 140 2465-2479
START_RANGE client2d socket172 150 2479-2494
START_RANGE client2e socket173 140 2494-2508
START_RANGE client2f socket174 150 2508-2523
START_RANGE client2g socket175 140 2523-2537
START_RANGE client2h socket176 150 2537-2552
START_RANGE client2i socket177 140 2552-2566
START_RANGE client2j socket178 150 2566-2581
START_RANGE client2k socket179 140 2581-2595
START_RANGE client2l socket180 150 2595-2610
START_RANGE client2m socket181 140 2610-2624
START_RANGE client2n socket182 150 2624-2639
START_RANGE client2a socket183 140 2639-2653
START_RANGE client2b socket184 150 2653-2668
START_RANGE client2c socket185 140 2668-2682
START_RANGE client2d socket186 150 2682-2697
START_RANGE client2e socket187 140 2697-2711
START_RANGE client2f socket188 150 2711-2726
START_RANGE client2g socket189 140 2726-2740
START_RANGE client2h socket190 150 2740-2755
START_RANGE client2i socket191 140 2755-2769
START_RANGE client2j socket192 150 2769-2784
START_RANGE client2k socket193 140 2784-2798
START_RANGE client2l socket194 150 2798-2813
START_RANGE client2m socket195 140 2813-2827
START_RANGE client2n socket196 150 2827-2842
START_RANGE client2a socket197 140 2842-2856
START_RANGE client2b socket198 150 2856-2871
START_RANGE client2c socket199 140 2871-2885
START_RANGE client2d socket200 150 2885-2900
START_RANGE client2e socket201 140 2900-2914
START_RANGE client2f socket202 150 2914-2929
START_RANGE client2g socket203 140 2929-2943
START_RANGE client2h socket204 150 2943-2958
START_RANGE client2i socket205 140 2958-2972
START_RANGE client2j socket206 150 2972-2987
START_RANGE client2k socket207 140 2987-3001
START_RANGE client2l socket208 150 3001-3016
START_RANGE client2m socket209 140 3016-3030
START_RANGE client2n socket210 150 3030-3045

START_RANGE client2o socket211 140 3045-3059
START_RANGE client2p socket212 150 3059-3074
START_RANGE client2q socket213 140 3074-3088
START_RANGE client2r socket214 150 3088-3103
START_RANGE client2s socket215 140 3103-3117
START_RANGE client2t socket216 150 3117-3132
START_RANGE client2u socket217 140 3132-3146
START_RANGE client2v socket218 150 3146-3161
START_RANGE client2w socket219 140 3161-3175
START_RANGE client2x socket220 150 3175-3190
START_RANGE client2y socket221 140 3190-3204
START_RANGE client2z socket222 150 3204-3219
START_RANGE client2aa socket223 140 3219-3233
START_RANGE client2ab socket224 150 3233-3248
START_RANGE client2o socket225 140 3248-3262
START_RANGE client2p socket226 150 3262-3277
START_RANGE client2q socket227 140 3277-3291
START_RANGE client2r socket228 150 3291-3306
START_RANGE client2s socket229 140 3306-3320
START_RANGE client2t socket230 150 3320-3335
START_RANGE client2u socket231 140 3335-3349
START_RANGE client2v socket232 150 3349-3364
START_RANGE client2w socket233 140 3364-3378
START_RANGE client2x socket234 150 3378-3393
START_RANGE client2y socket235 140 3393-3407
START_RANGE client2z socket236 150 3407-3422
START_RANGE client2aa socket237 140 3422-3436
START_RANGE client2ab socket238 150 3436-3451
START_RANGE client2o socket239 140 3451-3465
START_RANGE client2p socket240 150 3465-3480
START_RANGE client2q socket241 140 3480-3494
START_RANGE client2r socket242 150 3494-3509
START_RANGE client2s socket243 140 3509-3523
START_RANGE client2t socket244 150 3523-3538
START_RANGE client2u socket245 140 3538-3552
START_RANGE client2v socket246 150 3552-3567
START_RANGE client2w socket247 140 3567-3581
START_RANGE client2x socket248 150 3581-3596
START_RANGE client2y socket249 140 3596-3610
START_RANGE client2z socket250 150 3610-3625
START_RANGE client2aa socket251 140 3625-3639
START_RANGE client2ab socket252 150 3639-3654
START_RANGE client2o socket253 140 3654-3668
START_RANGE client2p socket254 150 3668-3683
START_RANGE client2q socket255 140 3683-3697
START_RANGE client2r socket256 150 3697-3712
START_RANGE client2s socket257 140 3712-3726
START_RANGE client2t socket258 150 3726-3741
START_RANGE client2u socket259 140 3741-3755
START_RANGE client2v socket260 150 3755-3770
START_RANGE client2w socket261 140 3770-3784
START_RANGE client2x socket262 150 3784-3799
START_RANGE client2y socket263 140 3799-3813
```

```
START_RANGE client2z socket264 150 3813-3828
START_RANGE client2aa socket265 140 3828-3842
START_RANGE client2ab socket266 150 3842-3857
START_RANGE client2o socket267 140 3857-3871
START_RANGE client2p socket268 150 3871-3886
START_RANGE client2q socket269 140 3886-3900
START_RANGE client2r socket270 150 3900-3915
START_RANGE client2s socket271 140 3915-3929
START_RANGE client2t socket272 150 3929-3944
START_RANGE client2u socket273 140 3944-3958
START_RANGE client2v socket274 150 3958-3973
START_RANGE client2w socket275 140 3973-3987
START_RANGE client2x socket276 150 3987-4002
START_RANGE client2y socket277 140 4002-4016
START_RANGE client2z socket278 150 4016-4031
START_RANGE client2aa socket279 140 4031-4045
START_RANGE client2ab socket280 150 4045-4060

START_RANGE client3a socket281 140 4060-4074
START_RANGE client3b socket282 150 4074-4089
START_RANGE client3c socket283 140 4089-4103
START_RANGE client3d socket284 150 4103-4118
START_RANGE client3e socket285 140 4118-4132
START_RANGE client3f socket286 150 4132-4147
START_RANGE client3g socket287 140 4147-4161
START_RANGE client3h socket288 150 4161-4176
START_RANGE client3i socket289 140 4176-4190
START_RANGE client3j socket290 150 4190-4205
START_RANGE client3k socket291 140 4205-4219
START_RANGE client3l socket292 150 4219-4234
START_RANGE client3m socket293 140 4234-4248
START_RANGE client3n socket294 150 4248-4263
START_RANGE client3a socket295 140 4263-4277
START_RANGE client3b socket296 150 4277-4292
START_RANGE client3c socket297 140 4292-4306
START_RANGE client3d socket298 150 4306-4321
START_RANGE client3e socket299 140 4321-4335
START_RANGE client3f socket300 150 4335-4350
START_RANGE client3g socket301 140 4350-4364
START_RANGE client3h socket302 150 4364-4379
START_RANGE client3i socket303 140 4379-4393
START_RANGE client3j socket304 150 4393-4408
START_RANGE client3k socket305 140 4408-4422
START_RANGE client3l socket306 150 4422-4437
START_RANGE client3m socket307 140 4437-4451
START_RANGE client3n socket308 150 4451-4466
START_RANGE client3a socket309 140 4466-4480
START_RANGE client3b socket310 150 4480-4495
START_RANGE client3c socket311 140 4495-4509
START_RANGE client3d socket312 150 4509-4524
START_RANGE client3e socket313 140 4524-4538
START_RANGE client3f socket314 150 4538-4553
START_RANGE client3g socket315 140 4553-4567
START_RANGE client3h socket316 150 4567-4582
START_RANGE client3i socket317 140 4582-4596
START_RANGE client3j socket318 150 4596-4611
START_RANGE client3k socket319 140 4611-4625
START_RANGE client3l socket320 150 4625-4640
START_RANGE client3m socket321 140 4640-4654
START_RANGE client3n socket322 150 4654-4669
START_RANGE client3a socket323 140 4669-4683
START_RANGE client3b socket324 150 4683-4698
START_RANGE client3c socket325 140 4698-4712
START_RANGE client3d socket326 150 4712-4727
START_RANGE client3e socket327 140 4727-4741
START_RANGE client3f socket328 150 4741-4756
START_RANGE client3g socket329 140 4756-4770
START_RANGE client3h socket330 150 4770-4785
START_RANGE client3i socket331 140 4785-4799
START_RANGE client3j socket332 150 4799-4814
START_RANGE client3k socket333 140 4814-4828
START_RANGE client3l socket334 150 4828-4843
START_RANGE client3m socket335 140 4843-4857
START_RANGE client3n socket336 150 4857-4872
START_RANGE client3a socket337 140 4872-4886
START_RANGE client3b socket338 150 4886-4901
START_RANGE client3c socket339 140 4901-4915
START_RANGE client3d socket340 150 4915-4930
START_RANGE client3e socket341 140 4930-4944
START_RANGE client3f socket342 150 4944-4959
START_RANGE client3g socket343 140 4959-4973
START_RANGE client3h socket344 150 4973-4988
START_RANGE client3i socket345 140 4988-5002
START_RANGE client3j socket346 150 5002-5017
START_RANGE client3k socket347 140 5017-5031
START_RANGE client3l socket348 150 5031-5046
START_RANGE client3m socket349 140 5046-5060
START_RANGE client3n socket350 150 5060-5075

START_RANGE client3o socket351 140 5075-5089
START_RANGE client3p socket352 150 5089-5104
START_RANGE client3q socket353 140 5104-5118
START_RANGE client3r socket354 150 5118-5133
START_RANGE client3s socket355 140 5133-5147
START_RANGE client3t socket356 150 5147-5162
START_RANGE client3u socket357 140 5162-5176
START_RANGE client3v socket358 150 5176-5191
START_RANGE client3w socket359 140 5191-5205
START_RANGE client3x socket360 150 5205-5220
START_RANGE client3y socket361 140 5220-5234
START_RANGE client3z socket362 150 5234-5249
START_RANGE client3aa socket363 140 5249-5263
START_RANGE client3ab socket364 150 5263-5278
START_RANGE client3o socket365 140 5278-5292
START_RANGE client3p socket366 150 5292-5307
START_RANGE client3q socket367 140 5307-5321
START_RANGE client3r socket368 150 5321-5336
START_RANGE client3s socket369 140 5336-5350
START_RANGE client3t socket370 150 5350-5365
START_RANGE client3u socket371 140 5365-5379
START_RANGE client3v socket372 150 5379-5394
START_RANGE client3w socket373 140 5394-5408
START_RANGE client3x socket374 150 5408-5423
START_RANGE client3y socket375 140 5423-5437
START_RANGE client3z socket376 150 5437-5452
START_RANGE client3aa socket377 140 5452-5466
START_RANGE client3ab socket378 150 5466-5481
START_RANGE client3o socket379 140 5481-5495
START_RANGE client3p socket380 150 5495-5510
START_RANGE client3q socket381 140 5510-5524
START_RANGE client3r socket382 150 5524-5539
START_RANGE client3s socket383 140 5539-5553
START_RANGE client3t socket384 150 5553-5568
START_RANGE client3u socket385 140 5568-5582
START_RANGE client3v socket386 150 5582-5597
START_RANGE client3w socket387 140 5597-5611
START_RANGE client3x socket388 150 5611-5626
START_RANGE client3y socket389 140 5626-5640
START_RANGE client3z socket390 150 5640-5655
START_RANGE client3aa socket391 140 5655-5669
START_RANGE client3ab socket392 150 5669-5684
START_RANGE client3o socket393 140 5684-5698
START_RANGE client3p socket394 150 5698-5713
START_RANGE client3q socket395 140 5713-5727
START_RANGE client3r socket396 150 5727-5742
START_RANGE client3s socket397 140 5742-5756
START_RANGE client3t socket398 150 5756-5771
START_RANGE client3u socket399 140 5771-5785
START_RANGE client3v socket400 150 5785-5800
START_RANGE client3w socket401 140 5800-5814
START_RANGE client3x socket402 150 5814-5829
START_RANGE client3y socket403 140 5829-5843
START_RANGE client3z socket404 150 5843-5858
START_RANGE client3aa socket405 140 5858-5872
START_RANGE client3ab socket406 150 5872-5887
START_RANGE client3o socket407 140 5887-5901
START_RANGE client3p socket408 150 5901-5916
START_RANGE client3q socket409 140 5916-5930
START_RANGE client3r socket410 150 5930-5945
START_RANGE client3s socket411 140 5945-5959
START_RANGE client3t socket412 150 5959-5974
START_RANGE client3u socket413 140 5974-5988
START_RANGE client3v socket414 150 5988-6003
START_RANGE client3w socket415 140 6003-6017
START_RANGE client3x socket416 150 6017-6032
START_RANGE client3y socket417 140 6032-6046
START_RANGE client3z socket418 150 6046-6061
START_RANGE client3aa socket419 140 6061-6075
START_RANGE client3ab socket420 150 6075-6090

#elif MASTER_NUM2
START_RANGE client4a socket421 140 6090-6104
START_RANGE client4b socket422 150 6104-6119
START_RANGE client4c socket423 140 6119-6133
START_RANGE client4d socket424 150 6133-6148
START_RANGE client4e socket425 140 6148-6162
START_RANGE client4f socket426 150 6162-6177
START_RANGE client4g socket427 140 6177-6191
START_RANGE client4h socket428 150 6191-6206
START_RANGE client4i socket429 140 6206-6220
START_RANGE client4j socket430 150 6220-6235
START_RANGE client4k socket431 140 6235-6249
START_RANGE client4l socket432 150 6249-6264
START_RANGE client4m socket433 140 6264-6278
START_RANGE client4n socket434 150 6278-6293
START_RANGE client4a socket435 140 6293-6307
START_RANGE client4b socket436 150 6307-6322
START_RANGE client4c socket437 140 6322-6336
START_RANGE client4d socket438 150 6336-6351
START_RANGE client4e socket439 140 6351-6365
START_RANGE client4f socket440 150 6365-6380
START_RANGE client4g socket441 140 6380-6394
START_RANGE client4h socket442 150 6394-6409
START_RANGE client4i socket443 140 6409-6423
START_RANGE client4j socket444 150 6423-6438
START_RANGE client4k socket445 140 6438-6452
START_RANGE client4l socket446 150 6452-6467
START_RANGE client4m socket447 140 6467-6481
START_RANGE client4n socket448 150 6481-6496
START_RANGE client4a socket449 140 6496-6510
START_RANGE client4b socket450 150 6510-6525
START_RANGE client4c socket451 140 6525-6539
START_RANGE client4d socket452 150 6539-6554
START_RANGE client4e socket453 140 6554-6568
START_RANGE client4f socket454 150 6568-6583
START_RANGE client4g socket455 140 6583-6597
START_RANGE client4h socket456 150 6597-6612
START_RANGE client4i socket457 140 6612-6626
```

```
START_RANGE client4j socket458 150 6626-6641
START_RANGE client4k socket459 140 6641-6655
START_RANGE client4l socket460 150 6655-6670
START_RANGE client4m socket461 140 6670-6684
START_RANGE client4n socket462 150 6684-6699
START_RANGE client4a socket463 140 6699-6713
START_RANGE client4b socket464 150 6713-6728
START_RANGE client4c socket465 140 6728-6742
START_RANGE client4d socket466 150 6742-6757
START_RANGE client4e socket467 140 6757-6771
START_RANGE client4f socket468 150 6771-6786
START_RANGE client4g socket469 140 6786-6800
START_RANGE client4h socket470 150 6800-6815
START_RANGE client4i socket471 140 6815-6829
START_RANGE client4j socket472 150 6829-6844
START_RANGE client4k socket473 140 6844-6858
START_RANGE client4l socket474 150 6858-6873
START_RANGE client4m socket475 140 6873-6887
START_RANGE client4n socket476 150 6887-6902
START_RANGE client4a socket477 140 6902-6916
START_RANGE client4b socket478 150 6916-6931
START_RANGE client4c socket479 140 6931-6945
START_RANGE client4d socket480 150 6945-6960
START_RANGE client4e socket481 140 6960-6974
START_RANGE client4f socket482 150 6974-6989
START_RANGE client4g socket483 140 6989-7003
START_RANGE client4h socket484 150 7003-7018
START_RANGE client4i socket485 140 7018-7032
START_RANGE client4j socket486 150 7032-7047
START_RANGE client4k socket487 140 7047-7061
START_RANGE client4l socket488 150 7061-7076
START_RANGE client4m socket489 140 7076-7090
START_RANGE client4n socket490 150 7090-7105

START_RANGE client4o socket491 140 7105-7119
START_RANGE client4p socket492 150 7119-7134
START_RANGE client4q socket493 140 7134-7148
START_RANGE client4r socket494 150 7148-7163
START_RANGE client4s socket495 140 7163-7177
START_RANGE client4t socket496 150 7177-7192
START_RANGE client4u socket497 140 7192-7206
START_RANGE client4v socket498 150 7206-7221
START_RANGE client4w socket499 140 7221-7235
START_RANGE client4x socket500 150 7235-7250
START_RANGE client4y socket501 140 7250-7264
START_RANGE client4z socket502 150 7264-7279
START_RANGE client4aa socket503 140 7279-7293
START_RANGE client4ab socket504 150 7293-7308
START_RANGE client4o socket505 140 7308-7322
START_RANGE client4p socket506 150 7322-7337
START_RANGE client4q socket507 140 7337-7351
START_RANGE client4r socket508 150 7351-7366
START_RANGE client4s socket509 140 7366-7380
START_RANGE client4t socket510 150 7380-7395
START_RANGE client4u socket511 140 7395-7409
START_RANGE client4v socket512 150 7409-7424
START_RANGE client4w socket513 140 7424-7438
START_RANGE client4x socket514 150 7438-7453
START_RANGE client4y socket515 140 7453-7467
START_RANGE client4z socket516 150 7467-7482
START_RANGE client4aa socket517 140 7482-7496
START_RANGE client4ab socket518 150 7496-7511
START_RANGE client4o socket519 140 7511-7525
START_RANGE client4p socket520 150 7525-7540
START_RANGE client4q socket521 140 7540-7554
START_RANGE client4r socket522 150 7554-7569
START_RANGE client4s socket523 140 7569-7583
START_RANGE client4t socket524 150 7583-7598
START_RANGE client4u socket525 140 7598-7612
START_RANGE client4v socket526 150 7612-7627
START_RANGE client4w socket527 140 7627-7641
START_RANGE client4x socket528 150 7641-7656
START_RANGE client4y socket529 140 7656-7670
START_RANGE client4z socket530 150 7670-7685
START_RANGE client4aa socket531 140 7685-7699
START_RANGE client4ab socket532 150 7699-7714
START_RANGE client4o socket533 140 7714-7728
START_RANGE client4p socket534 150 7728-7743
START_RANGE client4q socket535 140 7743-7757
START_RANGE client4r socket536 150 7757-7772
START_RANGE client4s socket537 140 7772-7786
START_RANGE client4t socket538 150 7786-7801
START_RANGE client4u socket539 140 7801-7815
START_RANGE client4v socket540 150 7815-7830
START_RANGE client4w socket541 140 7830-7844
START_RANGE client4x socket542 150 7844-7859
START_RANGE client4y socket543 140 7859-7873
START_RANGE client4z socket544 150 7873-7888
START_RANGE client4aa socket545 140 7888-7902
START_RANGE client4ab socket546 150 7902-7917
START_RANGE client4o socket547 140 7917-7931
START_RANGE client4p socket548 150 7931-7946
START_RANGE client4q socket549 140 7946-7960
START_RANGE client4r socket550 150 7960-7975
START_RANGE client4s socket551 140 7975-7989
START_RANGE client4t socket552 150 7989-8004
START_RANGE client4u socket553 140 8004-8018
START_RANGE client4v socket554 150 8018-8033
START_RANGE client4w socket555 140 8033-8047
START_RANGE client4x socket556 150 8047-8062
START_RANGE client4y socket557 140 8062-8076
START_RANGE client4z socket558 150 8076-8091
START_RANGE client4aa socket559 140 8091-8105
START_RANGE client4ab socket560 150 8105-8120

START_RANGE client5a socket561 140 8120-8134
START_RANGE client5b socket562 150 8134-8149
START_RANGE client5c socket563 140 8149-8163
START_RANGE client5d socket564 150 8163-8178
START_RANGE client5e socket565 140 8178-8192
START_RANGE client5f socket566 150 8192-8207
START_RANGE client5g socket567 140 8207-8221
START_RANGE client5h socket568 150 8221-8236
START_RANGE client5i socket569 140 8236-8250
START_RANGE client5j socket570 150 8250-8265
START_RANGE client5k socket571 140 8265-8279
START_RANGE client5l socket572 150 8279-8294
START_RANGE client5m socket573 140 8294-8308
START_RANGE client5n socket574 150 8308-8323
START_RANGE client5a socket575 140 8323-8337
START_RANGE client5b socket576 150 8337-8352
START_RANGE client5c socket577 140 8352-8366
START_RANGE client5d socket578 150 8366-8381
START_RANGE client5e socket579 140 8381-8395
START_RANGE client5f socket580 150 8395-8410
START_RANGE client5g socket581 140 8410-8424
START_RANGE client5h socket582 150 8424-8439
START_RANGE client5i socket583 140 8439-8453
START_RANGE client5j socket584 150 8453-8468
START_RANGE client5k socket585 140 8468-8482
START_RANGE client5l socket586 150 8482-8497
START_RANGE client5m socket587 140 8497-8511
START_RANGE client5n socket588 150 8511-8526
START_RANGE client5a socket589 140 8526-8540
START_RANGE client5b socket590 150 8540-8555
START_RANGE client5c socket591 140 8555-8569
START_RANGE client5d socket592 150 8569-8584
START_RANGE client5e socket593 140 8584-8598
START_RANGE client5f socket594 150 8598-8613
START_RANGE client5g socket595 140 8613-8627
START_RANGE client5h socket596 150 8627-8642
START_RANGE client5i socket597 140 8642-8656
START_RANGE client5j socket598 150 8656-8671
START_RANGE client5k socket599 140 8671-8685
START_RANGE client5l socket600 150 8685-8700
START_RANGE client5m socket601 140 8700-8714
START_RANGE client5n socket602 150 8714-8729
START_RANGE client5a socket603 140 8729-8743
START_RANGE client5b socket604 150 8743-8758
START_RANGE client5c socket605 140 8758-8772
START_RANGE client5d socket606 150 8772-8787
START_RANGE client5e socket607 140 8787-8801
START_RANGE client5f socket608 150 8801-8816
START_RANGE client5g socket609 140 8816-8830
START_RANGE client5h socket610 150 8830-8845
START_RANGE client5i socket611 140 8845-8859
START_RANGE client5j socket612 150 8859-8874
START_RANGE client5k socket613 140 8874-8888
START_RANGE client5l socket614 150 8888-8903
START_RANGE client5m socket615 140 8903-8917
START_RANGE client5n socket616 150 8917-8932
START_RANGE client5a socket617 140 8932-8946
START_RANGE client5b socket618 150 8946-8961
START_RANGE client5c socket619 140 8961-8975
START_RANGE client5d socket620 150 8975-8990
START_RANGE client5e socket621 140 8990-9004
START_RANGE client5f socket622 150 9004-9019
START_RANGE client5g socket623 140 9019-9033
START_RANGE client5h socket624 150 9033-9048
START_RANGE client5i socket625 140 9048-9062
START_RANGE client5j socket626 150 9062-9077
START_RANGE client5k socket627 140 9077-9091
START_RANGE client5l socket628 150 9091-9106
START_RANGE client5m socket629 140 9106-9120
START_RANGE client5n socket630 150 9120-9135

START_RANGE client5o socket631 140 9135-9149
START_RANGE client5p socket632 150 9149-9164
START_RANGE client5q socket633 140 9164-9178
START_RANGE client5r socket634 150 9178-9193
START_RANGE client5s socket635 140 9193-9207
START_RANGE client5t socket636 150 9207-9222
START_RANGE client5u socket637 140 9222-9236
START_RANGE client5v socket638 150 9236-9251
START_RANGE client5w socket639 140 9251-9265
START_RANGE client5x socket640 150 9265-9280
START_RANGE client5y socket641 140 9280-9294
START_RANGE client5z socket642 150 9294-9309
START_RANGE client5aa socket643 140 9309-9323
START_RANGE client5ab socket644 150 9323-9338
START_RANGE client5o socket645 140 9338-9352
START_RANGE client5p socket646 150 9352-9367
START_RANGE client5q socket647 140 9367-9381
START_RANGE client5r socket648 150 9381-9396
START_RANGE client5s socket649 140 9396-9410
START_RANGE client5t socket650 150 9410-9425
START_RANGE client5u socket651 140 9425-9439
START_RANGE client5v socket652 150 9439-9454
```

```
START_RANGE client5w socket653 140 9454-9468
START_RANGE client5x socket654 150 9468-9483
START_RANGE client5y socket655 140 9483-9497
START_RANGE client5z socket656 150 9497-9512
START_RANGE client5aa socket657 140 9512-9526
START_RANGE client5ab socket658 150 9526-9541
START_RANGE client5o socket659 140 9541-9555
START_RANGE client5p socket660 150 9555-9570
START_RANGE client5q socket661 140 9570-9584
START_RANGE client5r socket662 150 9584-9599
START_RANGE client5s socket663 140 9599-9613
START_RANGE client5t socket664 150 9613-9628
START_RANGE client5u socket665 140 9628-9642
START_RANGE client5v socket666 150 9642-9657
START_RANGE client5w socket667 140 9657-9671
START_RANGE client5x socket668 150 9671-9686
START_RANGE client5y socket669 140 9686-9700
START_RANGE client5z socket670 150 9700-9715
START_RANGE client5aa socket671 140 9715-9729
START_RANGE client5ab socket672 150 9729-9744
START_RANGE client5o socket673 140 9744-9758
START_RANGE client5p socket674 150 9758-9773
START_RANGE client5q socket675 140 9773-9787
START_RANGE client5r socket676 150 9787-9802
START_RANGE client5s socket677 140 9802-9816
START_RANGE client5t socket678 150 9816-9831
START_RANGE client5u socket679 140 9831-9845
START_RANGE client5v socket680 150 9845-9860
START_RANGE client5w socket681 140 9860-9874
START_RANGE client5x socket682 150 9874-9889
START_RANGE client5y socket683 140 9889-9903
START_RANGE client5z socket684 150 9903-9918
START_RANGE client5aa socket685 140 9918-9932
START_RANGE client5ab socket686 150 9932-9947
START_RANGE client5o socket687 140 9947-9961
START_RANGE client5p socket688 150 9961-9976
START_RANGE client5q socket689 140 9976-9990
START_RANGE client5r socket690 150 9990-10005
START_RANGE client5s socket691 140 10005-10019
START_RANGE client5t socket692 150 10019-10034
START_RANGE client5u socket693 140 10034-10048
START_RANGE client5v socket694 150 10048-10063
START_RANGE client5w socket695 140 10063-10077
START_RANGE client5x socket696 150 10077-10092
START_RANGE client5y socket697 140 10092-10106
START_RANGE client5z socket698 150 10106-10121
START_RANGE client5aa socket699 140 10121-10135
START_RANGE client5ab socket700 150 10135-10150

START_RANGE client6a socket701 140 10150-10164
START_RANGE client6b socket702 150 10164-10179
START_RANGE client6c socket703 140 10179-10193
START_RANGE client6d socket704 150 10193-10208
START_RANGE client6e socket705 140 10208-10222
START_RANGE client6f socket706 150 10222-10237
START_RANGE client6g socket707 140 10237-10251
START_RANGE client6h socket708 150 10251-10266
START_RANGE client6i socket709 140 10266-10280
START_RANGE client6j socket710 150 10280-10295
START_RANGE client6k socket711 140 10295-10309
START_RANGE client6l socket712 150 10309-10324
START_RANGE client6m socket713 140 10324-10338
START_RANGE client6n socket714 150 10338-10353
START_RANGE client6a socket715 140 10353-10367
START_RANGE client6b socket716 150 10367-10382
START_RANGE client6c socket717 140 10382-10396
START_RANGE client6d socket718 150 10396-10411
START_RANGE client6e socket719 140 10411-10425
START_RANGE client6f socket720 150 10425-10440
START_RANGE client6g socket721 140 10440-10454
START_RANGE client6h socket722 150 10454-10469
START_RANGE client6i socket723 140 10469-10483
START_RANGE client6j socket724 150 10483-10498
START_RANGE client6k socket725 140 10498-10512
START_RANGE client6l socket726 150 10512-10527
START_RANGE client6m socket727 140 10527-10541
START_RANGE client6n socket728 150 10541-10556
START_RANGE client6a socket729 140 10556-10570
START_RANGE client6b socket730 150 10570-10585
START_RANGE client6c socket731 140 10585-10599
START_RANGE client6d socket732 150 10599-10614
START_RANGE client6e socket733 140 10614-10628
START_RANGE client6f socket734 150 10628-10643
START_RANGE client6g socket735 140 10643-10657
START_RANGE client6h socket736 150 10657-10672
START_RANGE client6i socket737 140 10672-10686
START_RANGE client6j socket738 150 10686-10701
START_RANGE client6k socket739 140 10701-10715
START_RANGE client6l socket740 150 10715-10730
START_RANGE client6m socket741 140 10730-10744
START_RANGE client6n socket742 150 10744-10759
START_RANGE client6a socket743 140 10759-10773
START_RANGE client6b socket744 150 10773-10788
START_RANGE client6c socket745 140 10788-10802
START_RANGE client6d socket746 150 10802-10817
START_RANGE client6e socket747 140 10817-10831
START_RANGE client6f socket748 150 10831-10846
START_RANGE client6g socket749 140 10846-10860
START_RANGE client6h socket750 150 10860-10875
START_RANGE client6i socket751 140 10875-10889
START_RANGE client6j socket752 150 10889-10904
START_RANGE client6k socket753 140 10904-10918
START_RANGE client6l socket754 150 10918-10933
START_RANGE client6m socket755 140 10933-10947
START_RANGE client6n socket756 150 10947-10962
START_RANGE client6a socket757 140 10962-10976
START_RANGE client6b socket758 150 10976-10991
START_RANGE client6c socket759 140 10991-11005
START_RANGE client6d socket760 150 11005-11020
START_RANGE client6e socket761 140 11020-11034
START_RANGE client6f socket762 150 11034-11049
START_RANGE client6g socket763 140 11049-11063
START_RANGE client6h socket764 150 11063-11078
START_RANGE client6i socket765 140 11078-11092
START_RANGE client6j socket766 150 11092-11107
START_RANGE client6k socket767 140 11107-11121
START_RANGE client6l socket768 150 11121-11136
START_RANGE client6m socket769 140 11136-11150
START_RANGE client6n socket770 150 11150-11165

START_RANGE client6o socket771 140 11165-11179
START_RANGE client6p socket772 150 11179-11194
START_RANGE client6q socket773 140 11194-11208
START_RANGE client6r socket774 150 11208-11223
START_RANGE client6s socket775 140 11223-11237
START_RANGE client6t socket776 150 11237-11252
START_RANGE client6u socket777 140 11252-11266
START_RANGE client6v socket778 150 11266-11281
START_RANGE client6w socket779 140 11281-11295
START_RANGE client6x socket780 150 11295-11310
START_RANGE client6y socket781 140 11310-11324
START_RANGE client6z socket782 150 11324-11339
START_RANGE client6aa socket783 140 11339-11353
START_RANGE client6ab socket784 150 11353-11368
START_RANGE client6o socket785 140 11368-11382
START_RANGE client6p socket786 150 11382-11397
START_RANGE client6q socket787 140 11397-11411
START_RANGE client6r socket788 150 11411-11426
START_RANGE client6s socket789 140 11426-11440
START_RANGE client6t socket790 150 11440-11455
START_RANGE client6u socket791 140 11455-11469
START_RANGE client6v socket792 150 11469-11484
START_RANGE client6w socket793 140 11484-11498
START_RANGE client6x socket794 150 11498-11513
START_RANGE client6y socket795 140 11513-11527
START_RANGE client6z socket796 150 11527-11542
START_RANGE client6aa socket797 140 11542-11556
START_RANGE client6ab socket798 150 11556-11571
START_RANGE client6o socket799 140 11571-11585
START_RANGE client6p socket800 150 11585-11600
START_RANGE client6q socket801 140 11600-11614
START_RANGE client6r socket802 150 11614-11629
START_RANGE client6s socket803 140 11629-11643
START_RANGE client6t socket804 150 11643-11658
START_RANGE client6u socket805 140 11658-11672
START_RANGE client6v socket806 150 11672-11687
START_RANGE client6w socket807 140 11687-11701
START_RANGE client6x socket808 150 11701-11716
START_RANGE client6y socket809 140 11716-11730
START_RANGE client6z socket810 150 11730-11745
START_RANGE client6aa socket811 140 11745-11759
START_RANGE client6ab socket812 150 11759-11774
START_RANGE client6o socket813 140 11774-11788
START_RANGE client6p socket814 150 11788-11803
START_RANGE client6q socket815 140 11803-11817
START_RANGE client6r socket816 150 11817-11832
START_RANGE client6s socket817 140 11832-11846
START_RANGE client6t socket818 150 11846-11861
START_RANGE client6u socket819 140 11861-11875
START_RANGE client6v socket820 150 11875-11890
START_RANGE client6w socket821 140 11890-11904
START_RANGE client6x socket822 150 11904-11919
START_RANGE client6y socket823 140 11919-11933
START_RANGE client6z socket824 150 11933-11948
START_RANGE client6aa socket825 140 11948-11962
START_RANGE client6ab socket826 150 11962-11977
START_RANGE client6o socket827 140 11977-11991
START_RANGE client6p socket828 150 11991-12006
START_RANGE client6q socket829 140 12006-12020
START_RANGE client6r socket830 150 12020-12035
START_RANGE client6s socket831 140 12035-12049
START_RANGE client6t socket832 150 12049-12064
START_RANGE client6u socket833 140 12064-12078
START_RANGE client6v socket834 150 12078-12093
START_RANGE client6w socket835 140 12093-12107
START_RANGE client6x socket836 150 12107-12122
START_RANGE client6y socket837 140 12122-12136
START_RANGE client6z socket838 150 12136-12151
START_RANGE client6aa socket839 140 12151-12165
START_RANGE client6ab socket840 150 12165-12180

#elif MASTER_NUM3
START_RANGE client7a socket841 140 12180-12194
START_RANGE client7b socket842 150 12194-12209
START_RANGE client7c socket843 140 12209-12223
START_RANGE client7d socket844 150 12223-12238
START_RANGE client7e socket845 140 12238-12252
START_RANGE client7f socket846 150 12252-12267
```

```
START_RANGE client7g socket847 140 12267-12281
START_RANGE client7h socket848 150 12281-12296
START_RANGE client7i socket849 140 12296-12310
START_RANGE client7j socket850 150 12310-12325
START_RANGE client7k socket851 140 12325-12339
START_RANGE client7l socket852 150 12339-12354
START_RANGE client7m socket853 140 12354-12368
START_RANGE client7n socket854 150 12368-12383
START_RANGE client7a socket855 140 12383-12397
START_RANGE client7b socket856 150 12397-12412
START_RANGE client7c socket857 140 12412-12426
START_RANGE client7d socket858 150 12426-12441
START_RANGE client7e socket859 140 12441-12455
START_RANGE client7f socket860 150 12455-12470
START_RANGE client7g socket861 140 12470-12484
START_RANGE client7h socket862 150 12484-12499
START_RANGE client7i socket863 140 12499-12513
START_RANGE client7j socket864 150 12513-12528
START_RANGE client7k socket865 140 12528-12542
START_RANGE client7l socket866 150 12542-12557
START_RANGE client7m socket867 140 12557-12571
START_RANGE client7n socket868 150 12571-12586
START_RANGE client7a socket869 140 12586-12600
START_RANGE client7b socket870 150 12600-12615
START_RANGE client7c socket871 140 12615-12629
START_RANGE client7d socket872 150 12629-12644
START_RANGE client7e socket873 140 12644-12658
START_RANGE client7f socket874 150 12658-12673
START_RANGE client7g socket875 140 12673-12687
START_RANGE client7h socket876 150 12687-12702
START_RANGE client7i socket877 140 12702-12716
START_RANGE client7j socket878 150 12716-12731
START_RANGE client7k socket879 140 12731-12745
START_RANGE client7l socket880 150 12745-12760
START_RANGE client7m socket881 140 12760-12774
START_RANGE client7n socket882 150 12774-12789
START_RANGE client7a socket883 140 12789-12803
START_RANGE client7b socket884 150 12803-12818
START_RANGE client7c socket885 140 12818-12832
START_RANGE client7d socket886 150 12832-12847
START_RANGE client7e socket887 140 12847-12861
START_RANGE client7f socket888 150 12861-12876
START_RANGE client7g socket889 140 12876-12890
START_RANGE client7h socket890 150 12890-12905
START_RANGE client7i socket891 140 12905-12919
START_RANGE client7j socket892 150 12919-12934
START_RANGE client7k socket893 140 12934-12948
START_RANGE client7l socket894 150 12948-12963
START_RANGE client7m socket895 140 12963-12977
START_RANGE client7n socket896 150 12977-12992
START_RANGE client7a socket897 140 12992-13006
START_RANGE client7b socket898 150 13006-13021
START_RANGE client7c socket899 140 13021-13035
START_RANGE client7d socket900 150 13035-13050
START_RANGE client7e socket901 140 13050-13064
START_RANGE client7f socket902 150 13064-13079
START_RANGE client7g socket903 140 13079-13093
START_RANGE client7h socket904 150 13093-13108
START_RANGE client7i socket905 140 13108-13122
START_RANGE client7j socket906 150 13122-13137
START_RANGE client7k socket907 140 13137-13151
START_RANGE client7l socket908 150 13151-13166
START_RANGE client7m socket909 140 13166-13180
START_RANGE client7n socket910 150 13180-13195

START_RANGE client7o socket911 140 13195-13209
START_RANGE client7p socket912 150 13209-13224
START_RANGE client7q socket913 140 13224-13238
START_RANGE client7r socket914 150 13238-13253
START_RANGE client7s socket915 140 13253-13267
START_RANGE client7t socket916 150 13267-13282
START_RANGE client7u socket917 140 13282-13296
START_RANGE client7v socket918 150 13296-13311
START_RANGE client7w socket919 140 13311-13325
START_RANGE client7x socket920 150 13325-13340
START_RANGE client7y socket921 140 13340-13354
START_RANGE client7z socket922 150 13354-13369
START_RANGE client7aa socket923 140 13369-13383
START_RANGE client7ab socket924 150 13383-13398
START_RANGE client7o socket925 140 13398-13412
START_RANGE client7p socket926 150 13412-13427
START_RANGE client7q socket927 140 13427-13441
START_RANGE client7r socket928 150 13441-13456
START_RANGE client7s socket929 140 13456-13470
START_RANGE client7t socket930 150 13470-13485
START_RANGE client7u socket931 140 13485-13499
START_RANGE client7v socket932 150 13499-13514
START_RANGE client7w socket933 140 13514-13528
START_RANGE client7x socket934 150 13528-13543
START_RANGE client7y socket935 140 13543-13557
START_RANGE client7z socket936 150 13557-13572
START_RANGE client7aa socket937 140 13572-13586
START_RANGE client7ab socket938 150 13586-13601
START_RANGE client7o socket939 140 13601-13615
START_RANGE client7p socket940 150 13615-13630
START_RANGE client7q socket941 140 13630-13644
START_RANGE client7r socket942 150 13644-13659
START_RANGE client7s socket943 140 13659-13673
START_RANGE client7t socket944 150 13673-13688
START_RANGE client7u socket945 140 13688-13702
START_RANGE client7v socket946 150 13702-13717
START_RANGE client7w socket947 140 13717-13731
START_RANGE client7x socket948 150 13731-13746
START_RANGE client7y socket949 140 13746-13760
START_RANGE client7z socket950 150 13760-13775
START_RANGE client7aa socket951 140 13775-13789
START_RANGE client7ab socket952 150 13789-13804
START_RANGE client7o socket953 140 13804-13818
START_RANGE client7p socket954 150 13818-13833
START_RANGE client7q socket955 140 13833-13847
START_RANGE client7r socket956 150 13847-13862
START_RANGE client7s socket957 140 13862-13876
START_RANGE client7t socket958 150 13876-13891
START_RANGE client7u socket959 140 13891-13905
START_RANGE client7v socket960 150 13905-13920
START_RANGE client7w socket961 140 13920-13934
START_RANGE client7x socket962 150 13934-13949
START_RANGE client7y socket963 140 13949-13963
START_RANGE client7z socket964 150 13963-13978
START_RANGE client7aa socket965 140 13978-13992
START_RANGE client7ab socket966 150 13992-14007
START_RANGE client7o socket967 140 14007-14021
START_RANGE client7p socket968 150 14021-14036
START_RANGE client7q socket969 140 14036-14050
START_RANGE client7r socket970 150 14050-14065
START_RANGE client7s socket971 140 14065-14079
START_RANGE client7t socket972 150 14079-14094
START_RANGE client7u socket973 140 14094-14108
START_RANGE client7v socket974 150 14108-14123
START_RANGE client7w socket975 140 14123-14137
START_RANGE client7x socket976 150 14137-14152
START_RANGE client7y socket977 140 14152-14166
START_RANGE client7z socket978 150 14166-14181
START_RANGE client7aa socket979 140 14181-14195
START_RANGE client7ab socket980 150 14195-14210

START_RANGE client8a socket981 140 14210-14224
START_RANGE client8b socket982 150 14224-14239
START_RANGE client8c socket983 140 14239-14253
START_RANGE client8d socket984 150 14253-14268
START_RANGE client8e socket985 140 14268-14282
START_RANGE client8f socket986 150 14282-14297
START_RANGE client8g socket987 140 14297-14311
START_RANGE client8h socket988 150 14311-14326
START_RANGE client8i socket989 140 14326-14340
START_RANGE client8j socket990 150 14340-14355
START_RANGE client8k socket991 140 14355-14369
START_RANGE client8l socket992 150 14369-14384
START_RANGE client8m socket993 140 14384-14398
START_RANGE client8n socket994 150 14398-14413
START_RANGE client8a socket995 140 14413-14427
START_RANGE client8b socket996 150 14427-14442
START_RANGE client8c socket997 140 14442-14456
START_RANGE client8d socket998 150 14456-14471
START_RANGE client8e socket999 140 14471-14485
START_RANGE client8f socket1000 150 14485-14500
START_RANGE client8g socket1001 140 14500-14514
START_RANGE client8h socket1002 150 14514-14529
START_RANGE client8i socket1003 140 14529-14543
START_RANGE client8j socket1004 150 14543-14558
START_RANGE client8k socket1005 140 14558-14572
START_RANGE client8l socket1006 150 14572-14587
START_RANGE client8m socket1007 140 14587-14601
START_RANGE client8n socket1008 150 14601-14616
START_RANGE client8a socket1009 140 14616-14630
START_RANGE client8b socket1010 150 14630-14645
START_RANGE client8c socket1011 140 14645-14659
START_RANGE client8d socket1012 150 14659-14674
START_RANGE client8e socket1013 140 14674-14688
START_RANGE client8f socket1014 150 14688-14703
START_RANGE client8g socket1015 140 14703-14717
START_RANGE client8h socket1016 150 14717-14732
START_RANGE client8i socket1017 140 14732-14746
START_RANGE client8j socket1018 150 14746-14761
START_RANGE client8k socket1019 140 14761-14775
START_RANGE client8l socket1020 150 14775-14790
START_RANGE client8m socket1021 140 14790-14804
START_RANGE client8n socket1022 150 14804-14819
START_RANGE client8a socket1023 140 14819-14833
START_RANGE client8b socket1024 150 14833-14848
START_RANGE client8c socket1025 140 14848-14862
START_RANGE client8d socket1026 150 14862-14877
START_RANGE client8e socket1027 140 14877-14891
START_RANGE client8f socket1028 150 14891-14906
START_RANGE client8g socket1029 140 14906-14920
START_RANGE client8h socket1030 150 14920-14935
START_RANGE client8i socket1031 140 14935-14949
START_RANGE client8j socket1032 150 14949-14964
START_RANGE client8k socket1033 140 14964-14978
START_RANGE client8l socket1034 150 14978-14993
START_RANGE client8m socket1035 140 14993-15007
START_RANGE client8n socket1036 150 15007-15022
START_RANGE client8a socket1037 140 15022-15036
START_RANGE client8b socket1038 150 15036-15051
START_RANGE client8c socket1039 140 15051-15065
START_RANGE client8d socket1040 150 15065-15080
START_RANGE client8e socket1041 140 15080-15094
START_RANGE client8f socket1042 150 15094-15109
```

```
START_RANGE client8g socket1043 140 15109-15123
START_RANGE client8h socket1044 150 15123-15138
START_RANGE client8i socket1045 140 15138-15152
START_RANGE client8j socket1046 150 15152-15167
START_RANGE client8k socket1047 140 15167-15181
START_RANGE client8l socket1048 150 15181-15196
START_RANGE client8m socket1049 140 15196-15210
START_RANGE client8n socket1050 150 15210-15225

START_RANGE client8o socket1051 140 15225-15239
START_RANGE client8p socket1052 150 15239-15254
START_RANGE client8q socket1053 140 15254-15268
START_RANGE client8r socket1054 150 15268-15283
START_RANGE client8s socket1055 140 15283-15297
START_RANGE client8t socket1056 150 15297-15312
START_RANGE client8u socket1057 140 15312-15326
START_RANGE client8v socket1058 150 15326-15341
START_RANGE client8w socket1059 140 15341-15355
START_RANGE client8x socket1060 150 15355-15370
START_RANGE client8y socket1061 140 15370-15384
START_RANGE client8z socket1062 150 15384-15399
START_RANGE client8aa socket1063 140 15399-15413
START_RANGE client8ab socket1064 150 15413-15428
START_RANGE client8o socket1065 140 15428-15442
START_RANGE client8p socket1066 150 15442-15457
START_RANGE client8q socket1067 140 15457-15471
START_RANGE client8r socket1068 150 15471-15486
START_RANGE client8s socket1069 140 15486-15500
START_RANGE client8t socket1070 150 15500-15515
START_RANGE client8u socket1071 140 15515-15529
START_RANGE client8v socket1072 150 15529-15544
START_RANGE client8w socket1073 140 15544-15558
START_RANGE client8x socket1074 150 15558-15573
START_RANGE client8y socket1075 140 15573-15587
START_RANGE client8z socket1076 150 15587-15602
START_RANGE client8aa socket1077 140 15602-15616
START_RANGE client8ab socket1078 150 15616-15631
START_RANGE client8o socket1079 140 15631-15645
START_RANGE client8p socket1080 150 15645-15660
START_RANGE client8q socket1081 140 15660-15674
START_RANGE client8r socket1082 150 15674-15689
START_RANGE client8s socket1083 140 15689-15703
START_RANGE client8t socket1084 150 15703-15718
START_RANGE client8u socket1085 140 15718-15732
START_RANGE client8v socket1086 150 15732-15747
START_RANGE client8w socket1087 140 15747-15761
START_RANGE client8x socket1088 150 15761-15776
START_RANGE client8y socket1089 140 15776-15790
START_RANGE client8z socket1090 150 15790-15805
START_RANGE client8aa socket1091 140 15805-15819
START_RANGE client8ab socket1092 150 15819-15834
START_RANGE client8o socket1093 140 15834-15848
START_RANGE client8p socket1094 150 15848-15863
START_RANGE client8q socket1095 140 15863-15877
START_RANGE client8r socket1096 150 15877-15892
START_RANGE client8s socket1097 140 15892-15906
START_RANGE client8t socket1098 150 15906-15921
START_RANGE client8u socket1099 140 15921-15935
START_RANGE client8v socket1100 150 15935-15950
START_RANGE client8w socket1101 140 15950-15964
START_RANGE client8x socket1102 150 15964-15979
START_RANGE client8y socket1103 140 15979-15993
START_RANGE client8z socket1104 150 15993-16008
START_RANGE client8aa socket1105 140 16008-16022
START_RANGE client8ab socket1106 150 16022-16037
START_RANGE client8o socket1107 140 16037-16051
START_RANGE client8p socket1108 150 16051-16066
START_RANGE client8q socket1109 140 16066-16080
START_RANGE client8r socket1110 150 16080-16095
START_RANGE client8s socket1111 140 16095-16109
START_RANGE client8t socket1112 150 16109-16124
START_RANGE client8u socket1113 140 16124-16138
START_RANGE client8v socket1114 150 16138-16153
START_RANGE client8w socket1115 140 16153-16167
START_RANGE client8x socket1116 150 16167-16182
START_RANGE client8y socket1117 140 16182-16196
START_RANGE client8z socket1118 150 16196-16211
START_RANGE client8aa socket1119 140 16211-16225
START_RANGE client8ab socket1120 150 16225-16240

START_RANGE client9a socket1121 140 16240-16254
START_RANGE client9b socket1122 150 16254-16269
START_RANGE client9c socket1123 140 16269-16283
START_RANGE client9d socket1124 150 16283-16298
START_RANGE client9e socket1125 140 16298-16312
START_RANGE client9f socket1126 150 16312-16327
START_RANGE client9g socket1127 140 16327-16341
START_RANGE client9h socket1128 150 16341-16356
START_RANGE client9i socket1129 140 16356-16370
START_RANGE client9j socket1130 150 16370-16385
START_RANGE client9k socket1131 140 16385-16399
START_RANGE client9l socket1132 150 16399-16414
START_RANGE client9m socket1133 140 16414-16428
START_RANGE client9n socket1134 150 16428-16443
START_RANGE client9a socket1135 140 16443-16457
START_RANGE client9b socket1136 150 16457-16472
START_RANGE client9c socket1137 140 16472-16486
START_RANGE client9d socket1138 150 16486-16501
START_RANGE client9e socket1139 140 16501-16515
START_RANGE client9f socket1140 150 16515-16530
START_RANGE client9g socket1141 140 16530-16544
START_RANGE client9h socket1142 150 16544-16559
START_RANGE client9i socket1143 140 16559-16573
START_RANGE client9j socket1144 150 16573-16588
START_RANGE client9k socket1145 140 16588-16602
START_RANGE client9l socket1146 150 16602-16617
START_RANGE client9m socket1147 140 16617-16631
START_RANGE client9n socket1148 150 16631-16646
START_RANGE client9a socket1149 140 16646-16660
START_RANGE client9b socket1150 150 16660-16675
START_RANGE client9c socket1151 140 16675-16689
START_RANGE client9d socket1152 150 16689-16704
START_RANGE client9e socket1153 140 16704-16718
START_RANGE client9f socket1154 150 16718-16733
START_RANGE client9g socket1155 140 16733-16747
START_RANGE client9h socket1156 150 16747-16762
START_RANGE client9i socket1157 140 16762-16776
START_RANGE client9j socket1158 150 16776-16791
START_RANGE client9k socket1159 140 16791-16805
START_RANGE client9l socket1160 150 16805-16820
START_RANGE client9m socket1161 140 16820-16834
START_RANGE client9n socket1162 150 16834-16849
START_RANGE client9a socket1163 140 16849-16863
START_RANGE client9b socket1164 150 16863-16878
START_RANGE client9c socket1165 140 16878-16892
START_RANGE client9d socket1166 150 16892-16907
START_RANGE client9e socket1167 140 16907-16921
START_RANGE client9f socket1168 150 16921-16936
START_RANGE client9g socket1169 140 16936-16950
START_RANGE client9h socket1170 150 16950-16965
START_RANGE client9i socket1171 140 16965-16979
START_RANGE client9j socket1172 150 16979-16994
START_RANGE client9k socket1173 140 16994-17008
START_RANGE client9l socket1174 150 17008-17023
START_RANGE client9m socket1175 140 17023-17037
START_RANGE client9n socket1176 150 17037-17052
START_RANGE client9a socket1177 140 17052-17066
START_RANGE client9b socket1178 150 17066-17081
START_RANGE client9c socket1179 140 17081-17095
START_RANGE client9d socket1180 150 17095-17110
START_RANGE client9e socket1181 140 17110-17124
START_RANGE client9f socket1182 150 17124-17139
START_RANGE client9g socket1183 140 17139-17153
START_RANGE client9h socket1184 150 17153-17168
START_RANGE client9i socket1185 140 17168-17182
START_RANGE client9j socket1186 150 17182-17197
START_RANGE client9k socket1187 140 17197-17211
START_RANGE client9l socket1188 150 17211-17226
START_RANGE client9m socket1189 140 17226-17240
START_RANGE client9n socket1190 150 17240-17255

START_RANGE client9o socket1191 140 17255-17269
START_RANGE client9p socket1192 150 17269-17284
START_RANGE client9q socket1193 140 17284-17298
START_RANGE client9r socket1194 150 17298-17313
START_RANGE client9s socket1195 140 17313-17327
START_RANGE client9t socket1196 150 17327-17342
START_RANGE client9u socket1197 140 17342-17356
START_RANGE client9v socket1198 150 17356-17371
START_RANGE client9w socket1199 140 17371-17385
START_RANGE client9x socket1200 150 17385-17400
START_RANGE client9y socket1201 140 17400-17414
START_RANGE client9z socket1202 150 17414-17429
START_RANGE client9aa socket1203 140 17429-17443
START_RANGE client9ab socket1204 150 17443-17458
START_RANGE client9o socket1205 140 17458-17472
START_RANGE client9p socket1206 150 17472-17487
START_RANGE client9q socket1207 140 17487-17501
START_RANGE client9r socket1208 150 17501-17516
START_RANGE client9s socket1209 140 17516-17530
START_RANGE client9t socket1210 150 17530-17545
START_RANGE client9u socket1211 140 17545-17559
START_RANGE client9v socket1212 150 17559-17574
START_RANGE client9w socket1213 140 17574-17588
START_RANGE client9x socket1214 150 17588-17603
START_RANGE client9y socket1215 140 17603-17617
START_RANGE client9z socket1216 150 17617-17632
START_RANGE client9aa socket1217 140 17632-17646
START_RANGE client9ab socket1218 150 17646-17661
START_RANGE client9o socket1219 140 17661-17675
START_RANGE client9p socket1220 150 17675-17690
START_RANGE client9q socket1221 140 17690-17704
START_RANGE client9r socket1222 150 17704-17719
START_RANGE client9s socket1223 140 17719-17733
START_RANGE client9t socket1224 150 17733-17748
START_RANGE client9u socket1225 140 17748-17762
START_RANGE client9v socket1226 150 17762-17777
START_RANGE client9w socket1227 140 17777-17791
START_RANGE client9x socket1228 150 17791-17806
START_RANGE client9y socket1229 140 17806-17820
START_RANGE client9z socket1230 150 17820-17835
START_RANGE client9aa socket1231 140 17835-17849
START_RANGE client9ab socket1232 150 17849-17864
START_RANGE client9o socket1233 140 17864-17878
START_RANGE client9p socket1234 150 17878-17893
START_RANGE client9q socket1235 140 17893-17907
START_RANGE client9r socket1236 150 17907-17922
START_RANGE client9s socket1237 140 17922-17936
```

```
START_RANGE client9t socket1238 150 17936-17951
START_RANGE client9u socket1239 140 17951-17965
START_RANGE client9v socket1240 150 17965-17980
START_RANGE client9w socket1241 140 17980-17994
START_RANGE client9x socket1242 150 17994-18009
START_RANGE client9y socket1243 140 18009-18023
START_RANGE client9z socket1244 150 18023-18038
START_RANGE client9aa socket1245 140 18038-18052
START_RANGE client9ab socket1246 150 18052-18067
START_RANGE client9o socket1247 140 18067-18081
START_RANGE client9p socket1248 150 18081-18096
START_RANGE client9q socket1249 140 18096-18110
START_RANGE client9r socket1250 150 18110-18125
START_RANGE client9s socket1251 140 18125-18139
START_RANGE client9t socket1252 150 18139-18154
START_RANGE client9u socket1253 140 18154-18168
START_RANGE client9v socket1254 150 18168-18183
START_RANGE client9w socket1255 140 18183-18197
START_RANGE client9x socket1256 150 18197-18212
START_RANGE client9y socket1257 140 18212-18226
START_RANGE client9z socket1258 150 18226-18241
START_RANGE client9aa socket1259 140 18241-18255
START_RANGE client9ab socket1260 150 18255-18270

#elif MASTER_NUM4
START_RANGE client10a socket1261 140 18270-18284
START_RANGE client10b socket1262 150 18284-18299
START_RANGE client10c socket1263 140 18299-18313
START_RANGE client10d socket1264 150 18313-18328
START_RANGE client10e socket1265 140 18328-18342
START_RANGE client10f socket1266 150 18342-18357
START_RANGE client10g socket1267 140 18357-18371
START_RANGE client10h socket1268 150 18371-18386
START_RANGE client10i socket1269 140 18386-18400
START_RANGE client10j socket1270 150 18400-18415
START_RANGE client10k socket1271 140 18415-18429
START_RANGE client10l socket1272 150 18429-18444
START_RANGE client10m socket1273 140 18444-18458
START_RANGE client10n socket1274 150 18458-18473
START_RANGE client10a socket1275 140 18473-18487
START_RANGE client10b socket1276 150 18487-18502
START_RANGE client10c socket1277 140 18502-18516
START_RANGE client10d socket1278 150 18516-18531
START_RANGE client10e socket1279 140 18531-18545
START_RANGE client10f socket1280 150 18545-18560
START_RANGE client10g socket1281 140 18560-18574
START_RANGE client10h socket1282 150 18574-18589
START_RANGE client10i socket1283 140 18589-18603
START_RANGE client10j socket1284 150 18603-18618
START_RANGE client10k socket1285 140 18618-18632
START_RANGE client10l socket1286 150 18632-18647
START_RANGE client10m socket1287 140 18647-18661
START_RANGE client10n socket1288 150 18661-18676
START_RANGE client10a socket1289 140 18676-18690
START_RANGE client10b socket1290 150 18690-18705
START_RANGE client10c socket1291 140 18705-18719
START_RANGE client10d socket1292 150 18719-18734
START_RANGE client10e socket1293 140 18734-18748
START_RANGE client10f socket1294 150 18748-18763
START_RANGE client10g socket1295 140 18763-18777
START_RANGE client10h socket1296 150 18777-18792
START_RANGE client10i socket1297 140 18792-18806
START_RANGE client10j socket1298 150 18806-18821
START_RANGE client10k socket1299 140 18821-18835
START_RANGE client10l socket1300 150 18835-18850
START_RANGE client10m socket1301 140 18850-18864
START_RANGE client10n socket1302 150 18864-18879
START_RANGE client10a socket1303 140 18879-18893
START_RANGE client10b socket1304 150 18893-18908
START_RANGE client10c socket1305 140 18908-18922
START_RANGE client10d socket1306 150 18922-18937
START_RANGE client10e socket1307 140 18937-18951
START_RANGE client10f socket1308 150 18951-18966
START_RANGE client10g socket1309 140 18966-18980
START_RANGE client10h socket1310 150 18980-18995
START_RANGE client10i socket1311 140 18995-19009
START_RANGE client10j socket1312 150 19009-19024
START_RANGE client10k socket1313 140 19024-19038
START_RANGE client10l socket1314 150 19038-19053
START_RANGE client10m socket1315 140 19053-19067
START_RANGE client10n socket1316 150 19067-19082
START_RANGE client10a socket1317 140 19082-19096
START_RANGE client10b socket1318 150 19096-19111
START_RANGE client10c socket1319 140 19111-19125
START_RANGE client10d socket1320 150 19125-19140
START_RANGE client10e socket1321 140 19140-19154
START_RANGE client10f socket1322 150 19154-19169
START_RANGE client10g socket1323 140 19169-19183
START_RANGE client10h socket1324 150 19183-19198
START_RANGE client10i socket1325 140 19198-19212
START_RANGE client10j socket1326 150 19212-19227
START_RANGE client10k socket1327 140 19227-19241
START_RANGE client10l socket1328 150 19241-19256
START_RANGE client10m socket1329 140 19256-19270
START_RANGE client10n socket1330 150 19270-19285

START_RANGE client10o socket1331 140 19285-19299
START_RANGE client10p socket1332 150 19299-19314
START_RANGE client10q socket1333 140 19314-19328
START_RANGE client10r socket1334 150 19328-19343
START_RANGE client10s socket1335 140 19343-19357
START_RANGE client10t socket1336 150 19357-19372
START_RANGE client10u socket1337 140 19372-19386
START_RANGE client10v socket1338 150 19386-19401
START_RANGE client10w socket1339 140 19401-19415
START_RANGE client10x socket1340 150 19415-19430
START_RANGE client10y socket1341 140 19430-19444
START_RANGE client10z socket1342 150 19444-19459
START_RANGE client10aa socket1343 140 19459-19473
START_RANGE client10ab socket1344 150 19473-19488
START_RANGE client10o socket1345 140 19488-19502
START_RANGE client10p socket1346 150 19502-19517
START_RANGE client10q socket1347 140 19517-19531
START_RANGE client10r socket1348 150 19531-19546
START_RANGE client10s socket1349 140 19546-19560
START_RANGE client10t socket1350 150 19560-19575
START_RANGE client10u socket1351 140 19575-19589
START_RANGE client10v socket1352 150 19589-19604
START_RANGE client10w socket1353 140 19604-19618
START_RANGE client10x socket1354 150 19618-19633
START_RANGE client10y socket1355 140 19633-19647
START_RANGE client10z socket1356 150 19647-19662
START_RANGE client10aa socket1357 140 19662-19676
START_RANGE client10ab socket1358 150 19676-19691
START_RANGE client10o socket1359 140 19691-19705
START_RANGE client10p socket1360 150 19705-19720
START_RANGE client10q socket1361 140 19720-19734
START_RANGE client10r socket1362 150 19734-19749
START_RANGE client10s socket1363 140 19749-19763
START_RANGE client10t socket1364 150 19763-19778
START_RANGE client10u socket1365 140 19778-19792
START_RANGE client10v socket1366 150 19792-19807
START_RANGE client10w socket1367 140 19807-19821
START_RANGE client10x socket1368 150 19821-19836
START_RANGE client10y socket1369 140 19836-19850
START_RANGE client10z socket1370 150 19850-19865
START_RANGE client10aa socket1371 140 19865-19879
START_RANGE client10ab socket1372 150 19879-19894
START_RANGE client10o socket1373 140 19894-19908
START_RANGE client10p socket1374 150 19908-19923
START_RANGE client10q socket1375 140 19923-19937
START_RANGE client10r socket1376 150 19937-19952
START_RANGE client10s socket1377 140 19952-19966
START_RANGE client10t socket1378 150 19966-19981
START_RANGE client10u socket1379 140 19981-19995
START_RANGE client10v socket1380 150 19995-20010
START_RANGE client10w socket1381 140 20010-20024
START_RANGE client10x socket1382 150 20024-20039
START_RANGE client10y socket1383 140 20039-20053
START_RANGE client10z socket1384 150 20053-20068
START_RANGE client10aa socket1385 140 20068-20082
START_RANGE client10ab socket1386 150 20082-20097
START_RANGE client10o socket1387 140 20097-20111
START_RANGE client10p socket1388 150 20111-20126
START_RANGE client10q socket1389 140 20126-20140
START_RANGE client10r socket1390 150 20140-20155
START_RANGE client10s socket1391 140 20155-20169
START_RANGE client10t socket1392 150 20169-20184
START_RANGE client10u socket1393 140 20184-20198
START_RANGE client10v socket1394 150 20198-20213
START_RANGE client10w socket1395 140 20213-20227
START_RANGE client10x socket1396 150 20227-20242
START_RANGE client10y socket1397 140 20242-20256
START_RANGE client10z socket1398 150 20256-20271
START_RANGE client10aa socket1399 140 20271-20285
START_RANGE client10ab socket1400 150 20285-20300

START_RANGE client11a socket1401 140 20300-20314
START_RANGE client11b socket1402 150 20314-20329
START_RANGE client11c socket1403 140 20329-20343
START_RANGE client11d socket1404 150 20343-20358
START_RANGE client11e socket1405 140 20358-20372
START_RANGE client11f socket1406 150 20372-20387
START_RANGE client11g socket1407 140 20387-20401
START_RANGE client11h socket1408 150 20401-20416
START_RANGE client11i socket1409 140 20416-20430
START_RANGE client11j socket1410 150 20430-20445
START_RANGE client11k socket1411 140 20445-20459
START_RANGE client11l socket1412 150 20459-20474
START_RANGE client11m socket1413 140 20474-20488
START_RANGE client11n socket1414 150 20488-20503
START_RANGE client11a socket1415 140 20503-20517
START_RANGE client11b socket1416 150 20517-20532
START_RANGE client11c socket1417 140 20532-20546
START_RANGE client11d socket1418 150 20546-20561
START_RANGE client11e socket1419 140 20561-20575
START_RANGE client11f socket1420 150 20575-20590
START_RANGE client11g socket1421 140 20590-20604
START_RANGE client11h socket1422 150 20604-20619
START_RANGE client11i socket1423 140 20619-20633
START_RANGE client11j socket1424 150 20633-20648
START_RANGE client11k socket1425 140 20648-20662
START_RANGE client11l socket1426 150 20662-20677
START_RANGE client11m socket1427 140 20677-20691
START_RANGE client11n socket1428 150 20691-20706
START_RANGE client11a socket1429 140 20706-20720
START_RANGE client11b socket1430 150 20720-20735
START_RANGE client11c socket1431 140 20735-20749
```

```
START_RANGE client11d socket1432 150 20749-20764
START_RANGE client11e socket1433 140 20764-20778
START_RANGE client11f socket1434 150 20778-20793
START_RANGE client11g socket1435 140 20793-20807
START_RANGE client11h socket1436 150 20807-20822
START_RANGE client11i socket1437 140 20822-20836
START_RANGE client11j socket1438 150 20836-20851
START_RANGE client11k socket1439 140 20851-20865
START_RANGE client11l socket1440 150 20865-20880
START_RANGE client11m socket1441 140 20880-20894
START_RANGE client11n socket1442 150 20894-20909
START_RANGE client11a socket1443 140 20909-20923
START_RANGE client11b socket1444 150 20923-20938
START_RANGE client11c socket1445 140 20938-20952
START_RANGE client11d socket1446 150 20952-20967
START_RANGE client11e socket1447 140 20967-20981
START_RANGE client11f socket1448 150 20981-20996
START_RANGE client11g socket1449 140 20996-21010
START_RANGE client11h socket1450 150 21010-21025
START_RANGE client11i socket1451 140 21025-21039
START_RANGE client11j socket1452 150 21039-21054
START_RANGE client11k socket1453 140 21054-21068
START_RANGE client11l socket1454 150 21068-21083
START_RANGE client11m socket1455 140 21083-21097
START_RANGE client11n socket1456 150 21097-21112
START_RANGE client11a socket1457 140 21112-21126
START_RANGE client11b socket1458 150 21126-21141
START_RANGE client11c socket1459 140 21141-21155
START_RANGE client11d socket1460 150 21155-21170
START_RANGE client11e socket1461 140 21170-21184
START_RANGE client11f socket1462 150 21184-21199
START_RANGE client11g socket1463 140 21199-21213
START_RANGE client11h socket1464 150 21213-21228
START_RANGE client11i socket1465 140 21228-21242
START_RANGE client11j socket1466 150 21242-21257
START_RANGE client11k socket1467 140 21257-21271
START_RANGE client11l socket1468 150 21271-21286
START_RANGE client11m socket1469 140 21286-21300
START_RANGE client11n socket1470 150 21300-21315

START_RANGE client11o socket1471 140 21315-21329
START_RANGE client11p socket1472 150 21329-21344
START_RANGE client11q socket1473 140 21344-21358
START_RANGE client11r socket1474 150 21358-21373
START_RANGE client11s socket1475 140 21373-21387
START_RANGE client11t socket1476 150 21387-21402
START_RANGE client11u socket1477 140 21402-21416
START_RANGE client11v socket1478 150 21416-21431
START_RANGE client11w socket1479 140 21431-21445
START_RANGE client11x socket1480 150 21445-21460
START_RANGE client11y socket1481 140 21460-21474
START_RANGE client11z socket1482 150 21474-21489
START_RANGE client11aa socket1483 140 21489-21503
START_RANGE client11ab socket1484 150 21503-21518
START_RANGE client11o socket1485 140 21518-21532
START_RANGE client11p socket1486 150 21532-21547
START_RANGE client11q socket1487 140 21547-21561
START_RANGE client11r socket1488 150 21561-21576
START_RANGE client11s socket1489 140 21576-21590
START_RANGE client11t socket1490 150 21590-21605
START_RANGE client11u socket1491 140 21605-21619
START_RANGE client11v socket1492 150 21619-21634
START_RANGE client11w socket1493 140 21634-21648
START_RANGE client11x socket1494 150 21648-21663
START_RANGE client11y socket1495 140 21663-21677
START_RANGE client11z socket1496 150 21677-21692
START_RANGE client11aa socket1497 140 21692-21706
START_RANGE client11ab socket1498 150 21706-21721
START_RANGE client11o socket1499 140 21721-21735
START_RANGE client11p socket1500 150 21735-21750
START_RANGE client11q socket1501 140 21750-21764
START_RANGE client11r socket1502 150 21764-21779
START_RANGE client11s socket1503 140 21779-21793
START_RANGE client11t socket1504 150 21793-21808
START_RANGE client11u socket1505 140 21808-21822
START_RANGE client11v socket1506 150 21822-21837
START_RANGE client11w socket1507 140 21837-21851
START_RANGE client11x socket1508 150 21851-21866
START_RANGE client11y socket1509 140 21866-21880
START_RANGE client11z socket1510 150 21880-21895
START_RANGE client11aa socket1511 140 21895-21909
START_RANGE client11ab socket1512 150 21909-21924
START_RANGE client11o socket1513 140 21924-21938
START_RANGE client11p socket1514 150 21938-21953
START_RANGE client11q socket1515 140 21953-21967
START_RANGE client11r socket1516 150 21967-21982
START_RANGE client11s socket1517 140 21982-21996
START_RANGE client11t socket1518 150 21996-22011
START_RANGE client11u socket1519 140 22011-22025
START_RANGE client11v socket1520 150 22025-22040
START_RANGE client11w socket1521 140 22040-22054
START_RANGE client11x socket1522 150 22054-22069
START_RANGE client11y socket1523 140 22069-22083
START_RANGE client11z socket1524 150 22083-22098
START_RANGE client11aa socket1525 140 22098-22112
START_RANGE client11ab socket1526 150 22112-22127
START_RANGE client11o socket1527 140 22127-22141
START_RANGE client11p socket1528 150 22141-22156
START_RANGE client11q socket1529 140 22156-22170
START_RANGE client11r socket1530 150 22170-22185
START_RANGE client11s socket1531 140 22185-22199
START_RANGE client11t socket1532 150 22199-22214
START_RANGE client11u socket1533 140 22214-22228
START_RANGE client11v socket1534 150 22228-22243
START_RANGE client11w socket1535 140 22243-22257
START_RANGE client11x socket1536 150 22257-22272
START_RANGE client11y socket1537 140 22272-22286
START_RANGE client11z socket1538 150 22286-22301
START_RANGE client11aa socket1539 140 22301-22315
START_RANGE client11ab socket1540 150 22315-22330

START_RANGE client12a socket1541 140 22330-22344
START_RANGE client12b socket1542 150 22344-22359
START_RANGE client12c socket1543 140 22359-22373
START_RANGE client12d socket1544 150 22373-22388
START_RANGE client12e socket1545 140 22388-22402
START_RANGE client12f socket1546 150 22402-22417
START_RANGE client12g socket1547 140 22417-22431
START_RANGE client12h socket1548 150 22431-22446
START_RANGE client12i socket1549 140 22446-22460
START_RANGE client12j socket1550 150 22460-22475
START_RANGE client12k socket1551 140 22475-22489
START_RANGE client12l socket1552 150 22489-22504
START_RANGE client12m socket1553 140 22504-22518
START_RANGE client12n socket1554 150 22518-22533
START_RANGE client12a socket1555 140 22533-22547
START_RANGE client12b socket1556 150 22547-22562
START_RANGE client12c socket1557 140 22562-22576
START_RANGE client12d socket1558 150 22576-22591
START_RANGE client12e socket1559 140 22591-22605
START_RANGE client12f socket1560 150 22605-22620
START_RANGE client12g socket1561 140 22620-22634
START_RANGE client12h socket1562 150 22634-22649
START_RANGE client12i socket1563 140 22649-22663
START_RANGE client12j socket1564 150 22663-22678
START_RANGE client12k socket1565 140 22678-22692
START_RANGE client12l socket1566 150 22692-22707
START_RANGE client12m socket1567 140 22707-22721
START_RANGE client12n socket1568 150 22721-22736
START_RANGE client12a socket1569 140 22736-22750
START_RANGE client12b socket1570 150 22750-22765
START_RANGE client12c socket1571 140 22765-22779
START_RANGE client12d socket1572 150 22779-22794
START_RANGE client12e socket1573 140 22794-22808
START_RANGE client12f socket1574 150 22808-22823
START_RANGE client12g socket1575 140 22823-22837
START_RANGE client12h socket1576 150 22837-22852
START_RANGE client12i socket1577 140 22852-22866
START_RANGE client12j socket1578 150 22866-22881
START_RANGE client12k socket1579 140 22881-22895
START_RANGE client12l socket1580 150 22895-22910
START_RANGE client12m socket1581 140 22910-22924
START_RANGE client12n socket1582 150 22924-22939
START_RANGE client12a socket1583 140 22939-22953
START_RANGE client12b socket1584 150 22953-22968
START_RANGE client12c socket1585 140 22968-22982
START_RANGE client12d socket1586 150 22982-22997
START_RANGE client12e socket1587 140 22997-23011
START_RANGE client12f socket1588 150 23011-23026
START_RANGE client12g socket1589 140 23026-23040
START_RANGE client12h socket1590 150 23040-23055
START_RANGE client12i socket1591 140 23055-23069
START_RANGE client12j socket1592 150 23069-23084
START_RANGE client12k socket1593 140 23084-23098
START_RANGE client12l socket1594 150 23098-23113
START_RANGE client12m socket1595 140 23113-23127
START_RANGE client12n socket1596 150 23127-23142
START_RANGE client12a socket1597 140 23142-23156
START_RANGE client12b socket1598 150 23156-23171
START_RANGE client12c socket1599 140 23171-23185
START_RANGE client12d socket1600 150 23185-23200
START_RANGE client12e socket1601 140 23200-23214
START_RANGE client12f socket1602 150 23214-23229
START_RANGE client12g socket1603 140 23229-23243
START_RANGE client12h socket1604 150 23243-23258
START_RANGE client12i socket1605 140 23258-23272
START_RANGE client12j socket1606 150 23272-23287
START_RANGE client12k socket1607 140 23287-23301
START_RANGE client12l socket1608 150 23301-23316
START_RANGE client12m socket1609 140 23316-23330
START_RANGE client12n socket1610 150 23330-23345

START_RANGE client12o socket1611 140 23345-23359
START_RANGE client12p socket1612 150 23359-23374
START_RANGE client12q socket1613 140 23374-23388
START_RANGE client12r socket1614 150 23388-23403
START_RANGE client12s socket1615 140 23403-23417
START_RANGE client12t socket1616 150 23417-23432
START_RANGE client12u socket1617 140 23432-23446
START_RANGE client12v socket1618 150 23446-23461
START_RANGE client12w socket1619 140 23461-23475
START_RANGE client12x socket1620 150 23475-23490
START_RANGE client12y socket1621 140 23490-23504
START_RANGE client12z socket1622 150 23504-23519
START_RANGE client12aa socket1623 140 23519-23533
START_RANGE client12ab socket1624 150 23533-23548
START_RANGE client12o socket1625 140 23548-23562
START_RANGE client12p socket1626 150 23562-23577
```

```
START_RANGE client12q socket1627 140 23577-23591
START_RANGE client12r socket1628 150 23591-23606
START_RANGE client12s socket1629 140 23606-23620
START_RANGE client12t socket1630 150 23620-23635
START_RANGE client12u socket1631 140 23635-23649
START_RANGE client12v socket1632 150 23649-23664
START_RANGE client12w socket1633 140 23664-23678
START_RANGE client12x socket1634 150 23678-23693
START_RANGE client12y socket1635 140 23693-23707
START_RANGE client12z socket1636 150 23707-23722
START_RANGE client12aa socket1637 140 23722-23736
START_RANGE client12ab socket1638 150 23736-23751
START_RANGE client12o socket1639 140 23751-23765
START_RANGE client12p socket1640 150 23765-23780
START_RANGE client12q socket1641 140 23780-23794
START_RANGE client12r socket1642 150 23794-23809
START_RANGE client12s socket1643 140 23809-23823
START_RANGE client12t socket1644 150 23823-23838
START_RANGE client12u socket1645 140 23838-23852
START_RANGE client12v socket1646 150 23852-23867
START_RANGE client12w socket1647 140 23867-23881
START_RANGE client12x socket1648 150 23881-23896
START_RANGE client12y socket1649 140 23896-23910
START_RANGE client12z socket1650 150 23910-23925
START_RANGE client12aa socket1651 140 23925-23939
START_RANGE client12ab socket1652 150 23939-23954
START_RANGE client12o socket1653 140 23954-23968
START_RANGE client12p socket1654 150 23968-23983
START_RANGE client12q socket1655 140 23983-23997
START_RANGE client12r socket1656 150 23997-24012
START_RANGE client12s socket1657 140 24012-24026
START_RANGE client12t socket1658 150 24026-24041
START_RANGE client12u socket1659 140 24041-24055
START_RANGE client12v socket1660 150 24055-24070
START_RANGE client12w socket1661 140 24070-24084
START_RANGE client12x socket1662 150 24084-24099
START_RANGE client12y socket1663 140 24099-24113
START_RANGE client12z socket1664 150 24113-24128
START_RANGE client12aa socket1665 140 24128-24142
START_RANGE client12ab socket1666 150 24142-24157
START_RANGE client12o socket1667 140 24157-24171
START_RANGE client12p socket1668 150 24171-24186
START_RANGE client12q socket1669 140 24186-24200
START_RANGE client12r socket1670 150 24200-24215
START_RANGE client12s socket1671 140 24215-24229
START_RANGE client12t socket1672 150 24229-24244
START_RANGE client12u socket1673 140 24244-24258
START_RANGE client12v socket1674 150 24258-24273
START_RANGE client12w socket1675 140 24273-24287
START_RANGE client12x socket1676 150 24287-24302
START_RANGE client12y socket1677 140 24302-24316
START_RANGE client12z socket1678 150 24316-24331
START_RANGE client12aa socket1679 140 24331-24345
START_RANGE client12ab socket1680 150 24345-24360

#elif MASTER_NUM5
START_RANGE client13a socket1681 140 24360-24374
START_RANGE client13b socket1682 150 24374-24389
START_RANGE client13c socket1683 140 24389-24403
START_RANGE client13d socket1684 150 24403-24418
START_RANGE client13e socket1685 140 24418-24432
START_RANGE client13f socket1686 150 24432-24447
START_RANGE client13g socket1687 140 24447-24461
START_RANGE client13h socket1688 150 24461-24476
START_RANGE client13i socket1689 140 24476-24490
START_RANGE client13j socket1690 150 24490-24505
START_RANGE client13k socket1691 140 24505-24519
START_RANGE client13l socket1692 150 24519-24534
START_RANGE client13m socket1693 140 24534-24548
START_RANGE client13n socket1694 150 24548-24563
START_RANGE client13a socket1695 140 24563-24577
START_RANGE client13b socket1696 150 24577-24592
START_RANGE client13c socket1697 140 24592-24606
START_RANGE client13d socket1698 150 24606-24621
START_RANGE client13e socket1699 140 24621-24635
START_RANGE client13f socket1700 150 24635-24650
START_RANGE client13g socket1701 140 24650-24664
START_RANGE client13h socket1702 150 24664-24679
START_RANGE client13i socket1703 140 24679-24693
START_RANGE client13j socket1704 150 24693-24708
START_RANGE client13k socket1705 140 24708-24722
START_RANGE client13l socket1706 150 24722-24737
START_RANGE client13m socket1707 140 24737-24751
START_RANGE client13n socket1708 150 24751-24766
START_RANGE client13a socket1709 140 24766-24780
START_RANGE client13b socket1710 150 24780-24795
START_RANGE client13c socket1711 140 24795-24809
START_RANGE client13d socket1712 150 24809-24824
START_RANGE client13e socket1713 140 24824-24838
START_RANGE client13f socket1714 150 24838-24853
START_RANGE client13g socket1715 140 24853-24867
START_RANGE client13h socket1716 150 24867-24882
START_RANGE client13i socket1717 140 24882-24896
START_RANGE client13j socket1718 150 24896-24911
START_RANGE client13k socket1719 140 24911-24925
START_RANGE client13l socket1720 150 24925-24940
START_RANGE client13m socket1721 140 24940-24954
START_RANGE client13n socket1722 150 24954-24969
START_RANGE client13a socket1723 140 24969-24983
START_RANGE client13b socket1724 150 24983-24998
START_RANGE client13c socket1725 140 24998-25012
START_RANGE client13d socket1726 150 25012-25027
START_RANGE client13e socket1727 140 25027-25041
START_RANGE client13f socket1728 150 25041-25056
START_RANGE client13g socket1729 140 25056-25070
START_RANGE client13h socket1730 150 25070-25085
START_RANGE client13i socket1731 140 25085-25099
START_RANGE client13j socket1732 150 25099-25114
START_RANGE client13k socket1733 140 25114-25128
START_RANGE client13l socket1734 150 25128-25143
START_RANGE client13m socket1735 140 25143-25157
START_RANGE client13n socket1736 150 25157-25172
START_RANGE client13a socket1737 140 25172-25186
START_RANGE client13b socket1738 150 25186-25201
START_RANGE client13c socket1739 140 25201-25215
START_RANGE client13d socket1740 150 25215-25230
START_RANGE client13e socket1741 140 25230-25244
START_RANGE client13f socket1742 150 25244-25259
START_RANGE client13g socket1743 140 25259-25273
START_RANGE client13h socket1744 150 25273-25288
START_RANGE client13i socket1745 140 25288-25302
START_RANGE client13j socket1746 150 25302-25317
START_RANGE client13k socket1747 140 25317-25331
START_RANGE client13l socket1748 150 25331-25346
START_RANGE client13m socket1749 140 25346-25360
START_RANGE client13n socket1750 150 25360-25375

START_RANGE client13o socket1751 140 25375-25389
START_RANGE client13p socket1752 150 25389-25404
START_RANGE client13q socket1753 140 25404-25418
START_RANGE client13r socket1754 150 25418-25433
START_RANGE client13s socket1755 140 25433-25447
START_RANGE client13t socket1756 150 25447-25462
START_RANGE client13u socket1757 140 25462-25476
START_RANGE client13v socket1758 150 25476-25491
START_RANGE client13w socket1759 140 25491-25505
START_RANGE client13x socket1760 150 25505-25520
START_RANGE client13y socket1761 140 25520-25534
START_RANGE client13z socket1762 150 25534-25549
START_RANGE client13aa socket1763 140 25549-25563
START_RANGE client13ab socket1764 150 25563-25578
START_RANGE client13o socket1765 140 25578-25592
START_RANGE client13p socket1766 150 25592-25607
START_RANGE client13q socket1767 140 25607-25621
START_RANGE client13r socket1768 150 25621-25636
START_RANGE client13s socket1769 140 25636-25650
START_RANGE client13t socket1770 150 25650-25665
START_RANGE client13u socket1771 140 25665-25679
START_RANGE client13v socket1772 150 25679-25694
START_RANGE client13w socket1773 140 25694-25708
START_RANGE client13x socket1774 150 25708-25723
START_RANGE client13y socket1775 140 25723-25737
START_RANGE client13z socket1776 150 25737-25752
START_RANGE client13aa socket1777 140 25752-25766
START_RANGE client13ab socket1778 150 25766-25781
START_RANGE client13o socket1779 140 25781-25795
START_RANGE client13p socket1780 150 25795-25810
START_RANGE client13q socket1781 140 25810-25824
START_RANGE client13r socket1782 150 25824-25839
START_RANGE client13s socket1783 140 25839-25853
START_RANGE client13t socket1784 150 25853-25868
START_RANGE client13u socket1785 140 25868-25882
START_RANGE client13v socket1786 150 25882-25897
START_RANGE client13w socket1787 140 25897-25911
START_RANGE client13x socket1788 150 25911-25926
START_RANGE client13y socket1789 140 25926-25940
START_RANGE client13z socket1790 150 25940-25955
START_RANGE client13aa socket1791 140 25955-25969
START_RANGE client13ab socket1792 150 25969-25984
START_RANGE client13o socket1793 140 25984-25998
START_RANGE client13p socket1794 150 25998-26013
START_RANGE client13q socket1795 140 26013-26027
START_RANGE client13r socket1796 150 26027-26042
START_RANGE client13s socket1797 140 26042-26056
START_RANGE client13t socket1798 150 26056-26071
START_RANGE client13u socket1799 140 26071-26085
START_RANGE client13v socket1800 150 26085-26100
START_RANGE client13w socket1801 140 26100-26114
START_RANGE client13x socket1802 150 26114-26129
START_RANGE client13y socket1803 140 26129-26143
START_RANGE client13z socket1804 150 26143-26158
START_RANGE client13aa socket1805 140 26158-26172
START_RANGE client13ab socket1806 150 26172-26187
START_RANGE client13o socket1807 140 26187-26201
START_RANGE client13p socket1808 150 26201-26216
START_RANGE client13q socket1809 140 26216-26230
START_RANGE client13r socket1810 150 26230-26245
START_RANGE client13s socket1811 140 26245-26259
START_RANGE client13t socket1812 150 26259-26274
START_RANGE client13u socket1813 140 26274-26288
START_RANGE client13v socket1814 150 26288-26303
START_RANGE client13w socket1815 140 26303-26317
START_RANGE client13x socket1816 150 26317-26332
START_RANGE client13y socket1817 140 26332-26346
START_RANGE client13z socket1818 150 26346-26361
START_RANGE client13aa socket1819 140 26361-26375
START_RANGE client13ab socket1820 150 26375-26390
```

```
START_RANGE client14a socket1821 140 26390-26404
START_RANGE client14b socket1822 150 26404-26419
START_RANGE client14c socket1823 140 26419-26433
START_RANGE client14d socket1824 150 26433-26448
START_RANGE client14e socket1825 140 26448-26462
START_RANGE client14f socket1826 150 26462-26477
START_RANGE client14g socket1827 140 26477-26491
START_RANGE client14h socket1828 150 26491-26506
START_RANGE client14i socket1829 140 26506-26520
START_RANGE client14j socket1830 150 26520-26535
START_RANGE client14k socket1831 140 26535-26549
START_RANGE client14l socket1832 150 26549-26564
START_RANGE client14m socket1833 140 26564-26578
START_RANGE client14n socket1834 150 26578-26593
START_RANGE client14a socket1835 140 26593-26607
START_RANGE client14b socket1836 150 26607-26622
START_RANGE client14c socket1837 140 26622-26636
START_RANGE client14d socket1838 150 26636-26651
START_RANGE client14e socket1839 140 26651-26665
START_RANGE client14f socket1840 150 26665-26680
START_RANGE client14g socket1841 140 26680-26694
START_RANGE client14h socket1842 150 26694-26709
START_RANGE client14i socket1843 140 26709-26723
START_RANGE client14j socket1844 150 26723-26738
START_RANGE client14k socket1845 140 26738-26752
START_RANGE client14l socket1846 150 26752-26767
START_RANGE client14m socket1847 140 26767-26781
START_RANGE client14n socket1848 150 26781-26796
START_RANGE client14a socket1849 140 26796-26810
START_RANGE client14b socket1850 150 26810-26825
START_RANGE client14c socket1851 140 26825-26839
START_RANGE client14d socket1852 150 26839-26854
START_RANGE client14e socket1853 140 26854-26868
START_RANGE client14f socket1854 150 26868-26883
START_RANGE client14g socket1855 140 26883-26897
START_RANGE client14h socket1856 150 26897-26912
START_RANGE client14i socket1857 140 26912-26926
START_RANGE client14j socket1858 150 26926-26941
START_RANGE client14k socket1859 140 26941-26955
START_RANGE client14l socket1860 150 26955-26970
START_RANGE client14m socket1861 140 26970-26984
START_RANGE client14n socket1862 150 26984-26999
START_RANGE client14a socket1863 140 26999-27013
START_RANGE client14b socket1864 150 27013-27028
START_RANGE client14c socket1865 140 27028-27042
START_RANGE client14d socket1866 150 27042-27057
START_RANGE client14e socket1867 140 27057-27071
START_RANGE client14f socket1868 150 27071-27086
START_RANGE client14g socket1869 140 27086-27100
START_RANGE client14h socket1870 150 27100-27115
START_RANGE client14i socket1871 140 27115-27129
START_RANGE client14j socket1872 150 27129-27144
START_RANGE client14k socket1873 140 27144-27158
START_RANGE client14l socket1874 150 27158-27173
START_RANGE client14m socket1875 140 27173-27187
START_RANGE client14n socket1876 150 27187-27202
START_RANGE client14a socket1877 140 27202-27216
START_RANGE client14b socket1878 150 27216-27231
START_RANGE client14c socket1879 140 27231-27245
START_RANGE client14d socket1880 150 27245-27260
START_RANGE client14e socket1881 140 27260-27274
START_RANGE client14f socket1882 150 27274-27289
START_RANGE client14g socket1883 140 27289-27303
START_RANGE client14h socket1884 150 27303-27318
START_RANGE client14i socket1885 140 27318-27332
START_RANGE client14j socket1886 150 27332-27347
START_RANGE client14k socket1887 140 27347-27361
START_RANGE client14l socket1888 150 27361-27376
START_RANGE client14m socket1889 140 27376-27390
START_RANGE client14n socket1890 150 27390-27405

START_RANGE client14o socket1891 140 27405-27419
START_RANGE client14p socket1892 150 27419-27434
START_RANGE client14q socket1893 140 27434-27448
START_RANGE client14r socket1894 150 27448-27463
START_RANGE client14s socket1895 140 27463-27477
START_RANGE client14t socket1896 150 27477-27492
START_RANGE client14u socket1897 140 27492-27506
START_RANGE client14v socket1898 150 27506-27521
START_RANGE client14w socket1899 140 27521-27535
START_RANGE client14x socket1900 150 27535-27550
START_RANGE client14y socket1901 140 27550-27564
START_RANGE client14z socket1902 150 27564-27579
START_RANGE client14aa socket1903 140 27579-27593
START_RANGE client14ab socket1904 150 27593-27608
START_RANGE client14o socket1905 140 27608-27622
START_RANGE client14p socket1906 150 27622-27637
START_RANGE client14q socket1907 140 27637-27651
START_RANGE client14r socket1908 150 27651-27666
START_RANGE client14s socket1909 140 27666-27680
START_RANGE client14t socket1910 150 27680-27695
START_RANGE client14u socket1911 140 27695-27709
START_RANGE client14v socket1912 150 27709-27724
START_RANGE client14w socket1913 140 27724-27738
START_RANGE client14x socket1914 150 27738-27753
START_RANGE client14y socket1915 140 27753-27767
START_RANGE client14z socket1916 150 27767-27782
START_RANGE client14aa socket1917 140 27782-27796
START_RANGE client14ab socket1918 150 27796-27811
START_RANGE client14o socket1919 140 27811-27825
START_RANGE client14p socket1920 150 27825-27840
START_RANGE client14q socket1921 140 27840-27854
START_RANGE client14r socket1922 150 27854-27869
START_RANGE client14s socket1923 140 27869-27883
START_RANGE client14t socket1924 150 27883-27898
START_RANGE client14u socket1925 140 27898-27912
START_RANGE client14v socket1926 150 27912-27927
START_RANGE client14w socket1927 140 27927-27941
START_RANGE client14x socket1928 150 27941-27956
START_RANGE client14y socket1929 140 27956-27970
START_RANGE client14z socket1930 150 27970-27985
START_RANGE client14aa socket1931 140 27985-27999
START_RANGE client14ab socket1932 150 27999-28014
START_RANGE client14o socket1933 140 28014-28028
START_RANGE client14p socket1934 150 28028-28043
START_RANGE client14q socket1935 140 28043-28057
START_RANGE client14r socket1936 150 28057-28072
START_RANGE client14s socket1937 140 28072-28086
START_RANGE client14t socket1938 150 28086-28101
START_RANGE client14u socket1939 140 28101-28115
START_RANGE client14v socket1940 150 28115-28130
START_RANGE client14w socket1941 140 28130-28144
START_RANGE client14x socket1942 150 28144-28159
START_RANGE client14y socket1943 140 28159-28173
START_RANGE client14z socket1944 150 28173-28188
START_RANGE client14aa socket1945 140 28188-28202
START_RANGE client14ab socket1946 150 28202-28217
START_RANGE client14o socket1947 140 28217-28231
START_RANGE client14p socket1948 150 28231-28246
START_RANGE client14q socket1949 140 28246-28260
START_RANGE client14r socket1950 150 28260-28275
START_RANGE client14s socket1951 140 28275-28289
START_RANGE client14t socket1952 150 28289-28304
START_RANGE client14u socket1953 140 28304-28318
START_RANGE client14v socket1954 150 28318-28333
START_RANGE client14w socket1955 140 28333-28347
START_RANGE client14x socket1956 150 28347-28362
START_RANGE client14y socket1957 140 28362-28376
START_RANGE client14z socket1958 150 28376-28391
START_RANGE client14aa socket1959 140 28391-28405
START_RANGE client14ab socket1960 150 28405-28420

START_RANGE client15a socket1961 140 28420-28434
START_RANGE client15b socket1962 150 28434-28449
START_RANGE client15c socket1963 140 28449-28463
START_RANGE client15d socket1964 150 28463-28478
START_RANGE client15e socket1965 140 28478-28492
START_RANGE client15f socket1966 150 28492-28507
START_RANGE client15g socket1967 140 28507-28521
START_RANGE client15h socket1968 150 28521-28536
START_RANGE client15i socket1969 140 28536-28550
START_RANGE client15j socket1970 150 28550-28565
START_RANGE client15k socket1971 140 28565-28579
START_RANGE client15l socket1972 150 28579-28594
START_RANGE client15m socket1973 140 28594-28608
START_RANGE client15n socket1974 150 28608-28623
START_RANGE client15a socket1975 140 28623-28637
START_RANGE client15b socket1976 150 28637-28652
START_RANGE client15c socket1977 140 28652-28666
START_RANGE client15d socket1978 150 28666-28681
START_RANGE client15e socket1979 140 28681-28695
START_RANGE client15f socket1980 150 28695-28710
START_RANGE client15g socket1981 140 28710-28724
START_RANGE client15h socket1982 150 28724-28739
START_RANGE client15i socket1983 140 28739-28753
START_RANGE client15j socket1984 150 28753-28768
START_RANGE client15k socket1985 140 28768-28782
START_RANGE client15l socket1986 150 28782-28797
START_RANGE client15m socket1987 140 28797-28811
START_RANGE client15n socket1988 150 28811-28826
START_RANGE client15a socket1989 140 28826-28840
START_RANGE client15b socket1990 150 28840-28855
START_RANGE client15c socket1991 140 28855-28869
START_RANGE client15d socket1992 150 28869-28884
START_RANGE client15e socket1993 140 28884-28898
START_RANGE client15f socket1994 150 28898-28913
START_RANGE client15g socket1995 140 28913-28927
START_RANGE client15h socket1996 150 28927-28942
START_RANGE client15i socket1997 140 28942-28956
START_RANGE client15j socket1998 150 28956-28971
START_RANGE client15k socket1999 140 28971-28985
START_RANGE client15l socket2000 150 28985-29000
START_RANGE client15m socket2001 140 29000-29014
START_RANGE client15n socket2002 150 29014-29029
START_RANGE client15a socket2003 140 29029-29043
START_RANGE client15b socket2004 150 29043-29058
START_RANGE client15c socket2005 140 29058-29072
START_RANGE client15d socket2006 150 29072-29087
START_RANGE client15e socket2007 140 29087-29101
START_RANGE client15f socket2008 150 29101-29116
START_RANGE client15g socket2009 140 29116-29130
START_RANGE client15h socket2010 150 29130-29145
START_RANGE client15i socket2011 140 29145-29159
START_RANGE client15j socket2012 150 29159-29174
START_RANGE client15k socket2013 140 29174-29188
START_RANGE client15l socket2014 150 29188-29203
START_RANGE client15m socket2015 140 29203-29217
```

```
START_RANGE client15n socket2016 150 29217-29232
START_RANGE client15a socket2017 140 29232-29246
START_RANGE client15b socket2018 150 29246-29261
START_RANGE client15c socket2019 140 29261-29275
START_RANGE client15d socket2020 150 29275-29290
START_RANGE client15e socket2021 140 29290-29304
START_RANGE client15f socket2022 150 29304-29319
START_RANGE client15g socket2023 140 29319-29333
START_RANGE client15h socket2024 150 29333-29348
START_RANGE client15i socket2025 140 29348-29362
START_RANGE client15j socket2026 150 29362-29377
START_RANGE client15k socket2027 140 29377-29391
START_RANGE client15l socket2028 150 29391-29406
START_RANGE client15m socket2029 140 29406-29420
START_RANGE client15n socket2030 150 29420-29435

START_RANGE client15o socket2031 140 29435-29449
START_RANGE client15p socket2032 150 29449-29464
START_RANGE client15q socket2033 140 29464-29478
START_RANGE client15r socket2034 150 29478-29493
START_RANGE client15s socket2035 140 29493-29507
START_RANGE client15t socket2036 150 29507-29522
START_RANGE client15u socket2037 140 29522-29536
START_RANGE client15v socket2038 150 29536-29551
START_RANGE client15w socket2039 140 29551-29565
START_RANGE client15x socket2040 150 29565-29580
START_RANGE client15y socket2041 140 29580-29594
START_RANGE client15z socket2042 150 29594-29609
START_RANGE client15aa socket2043 140 29609-29623
START_RANGE client15ab socket2044 150 29623-29638
START_RANGE client15o socket2045 140 29638-29652
START_RANGE client15p socket2046 150 29652-29667
START_RANGE client15q socket2047 140 29667-29681
START_RANGE client15r socket2048 150 29681-29696
START_RANGE client15s socket2049 140 29696-29710
START_RANGE client15t socket2050 150 29710-29725
START_RANGE client15u socket2051 140 29725-29739
START_RANGE client15v socket2052 150 29739-29754
START_RANGE client15w socket2053 140 29754-29768
START_RANGE client15x socket2054 150 29768-29783
START_RANGE client15y socket2055 140 29783-29797
START_RANGE client15z socket2056 150 29797-29812
START_RANGE client15aa socket2057 140 29812-29826
START_RANGE client15ab socket2058 150 29826-29841
START_RANGE client15o socket2059 140 29841-29855
START_RANGE client15p socket2060 150 29855-29870
START_RANGE client15q socket2061 140 29870-29884
START_RANGE client15r socket2062 150 29884-29899
START_RANGE client15s socket2063 140 29899-29913
START_RANGE client15t socket2064 150 29913-29928
START_RANGE client15u socket2065 140 29928-29942
START_RANGE client15v socket2066 150 29942-29957
START_RANGE client15w socket2067 140 29957-29971
START_RANGE client15x socket2068 150 29971-29986
START_RANGE client15y socket2069 140 29986-30000
START_RANGE client15z socket2070 150 30000-30015
START_RANGE client15aa socket2071 140 30015-30029
START_RANGE client15ab socket2072 150 30029-30044
START_RANGE client15o socket2073 140 30044-30058
START_RANGE client15p socket2074 150 30058-30073
START_RANGE client15q socket2075 140 30073-30087
START_RANGE client15r socket2076 150 30087-30102
START_RANGE client15s socket2077 140 30102-30116
START_RANGE client15t socket2078 150 30116-30131
START_RANGE client15u socket2079 140 30131-30145
START_RANGE client15v socket2080 150 30145-30160
START_RANGE client15w socket2081 140 30160-30174
START_RANGE client15x socket2082 150 30174-30189
START_RANGE client15y socket2083 140 30189-30203
START_RANGE client15z socket2084 150 30203-30218
START_RANGE client15aa socket2085 140 30218-30232
START_RANGE client15ab socket2086 150 30232-30247
START_RANGE client15o socket2087 140 30247-30261
START_RANGE client15p socket2088 150 30261-30276
START_RANGE client15q socket2089 140 30276-30290
START_RANGE client15r socket2090 150 30290-30305
START_RANGE client15s socket2091 140 30305-30319
START_RANGE client15t socket2092 150 30319-30334
START_RANGE client15u socket2093 140 30334-30348
START_RANGE client15v socket2094 150 30348-30363
START_RANGE client15w socket2095 140 30363-30377
START_RANGE client15x socket2096 150 30377-30392
START_RANGE client15y socket2097 140 30392-30406
START_RANGE client15z socket2098 150 30406-30421
START_RANGE client15aa socket2099 140 30421-30435
START_RANGE client15ab socket2100 150 30435-30450

#elif MASTER_NUM6
START_RANGE client16a socket2101 140 30450-30464
START_RANGE client16b socket2102 150 30464-30479
START_RANGE client16c socket2103 140 30479-30493
START_RANGE client16d socket2104 150 30493-30508
START_RANGE client16e socket2105 140 30508-30522
START_RANGE client16f socket2106 150 30522-30537
START_RANGE client16g socket2107 140 30537-30551
START_RANGE client16h socket2108 150 30551-30566
START_RANGE client16i socket2109 140 30566-30580
START_RANGE client16j socket2110 150 30580-30595
START_RANGE client16k socket2111 140 30595-30609
START_RANGE client16l socket2112 150 30609-30624
START_RANGE client16m socket2113 140 30624-30638
START_RANGE client16n socket2114 150 30638-30653
START_RANGE client16a socket2115 140 30653-30667
START_RANGE client16b socket2116 150 30667-30682
START_RANGE client16c socket2117 140 30682-30696
START_RANGE client16d socket2118 150 30696-30711
START_RANGE client16e socket2119 140 30711-30725
START_RANGE client16f socket2120 150 30725-30740
START_RANGE client16g socket2121 140 30740-30754
START_RANGE client16h socket2122 150 30754-30769
START_RANGE client16i socket2123 140 30769-30783
START_RANGE client16j socket2124 150 30783-30798
START_RANGE client16k socket2125 140 30798-30812
START_RANGE client16l socket2126 150 30812-30827
START_RANGE client16m socket2127 140 30827-30841
START_RANGE client16n socket2128 150 30841-30856
START_RANGE client16a socket2129 140 30856-30870
START_RANGE client16b socket2130 150 30870-30885
START_RANGE client16c socket2131 140 30885-30899
START_RANGE client16d socket2132 150 30899-30914
START_RANGE client16e socket2133 140 30914-30928
START_RANGE client16f socket2134 150 30928-30943
START_RANGE client16g socket2135 140 30943-30957
START_RANGE client16h socket2136 150 30957-30972
START_RANGE client16i socket2137 140 30972-30986
START_RANGE client16j socket2138 150 30986-31001
START_RANGE client16k socket2139 140 31001-31015
START_RANGE client16l socket2140 150 31015-31030
START_RANGE client16m socket2141 140 31030-31044
START_RANGE client16n socket2142 150 31044-31059
START_RANGE client16a socket2143 140 31059-31073
START_RANGE client16b socket2144 150 31073-31088
START_RANGE client16c socket2145 140 31088-31102
START_RANGE client16d socket2146 150 31102-31117
START_RANGE client16e socket2147 140 31117-31131
START_RANGE client16f socket2148 150 31131-31146
START_RANGE client16g socket2149 140 31146-31160
START_RANGE client16h socket2150 150 31160-31175
START_RANGE client16i socket2151 140 31175-31189
START_RANGE client16j socket2152 150 31189-31204
START_RANGE client16k socket2153 140 31204-31218
START_RANGE client16l socket2154 150 31218-31233
START_RANGE client16m socket2155 140 31233-31247
START_RANGE client16n socket2156 150 31247-31262
START_RANGE client16a socket2157 140 31262-31276
START_RANGE client16b socket2158 150 31276-31291
START_RANGE client16c socket2159 140 31291-31305
START_RANGE client16d socket2160 150 31305-31320
START_RANGE client16e socket2161 140 31320-31334
START_RANGE client16f socket2162 150 31334-31349
START_RANGE client16g socket2163 140 31349-31363
START_RANGE client16h socket2164 150 31363-31378
START_RANGE client16i socket2165 140 31378-31392
START_RANGE client16j socket2166 150 31392-31407
START_RANGE client16k socket2167 140 31407-31421
START_RANGE client16l socket2168 150 31421-31436
START_RANGE client16m socket2169 140 31436-31450
START_RANGE client16n socket2170 150 31450-31465

START_RANGE client16o socket2171 140 31465-31479
START_RANGE client16p socket2172 150 31479-31494
START_RANGE client16q socket2173 140 31494-31508
START_RANGE client16r socket2174 150 31508-31523
START_RANGE client16s socket2175 140 31523-31537
START_RANGE client16t socket2176 150 31537-31552
START_RANGE client16u socket2177 140 31552-31566
START_RANGE client16v socket2178 150 31566-31581
START_RANGE client16w socket2179 140 31581-31595
START_RANGE client16x socket2180 150 31595-31610
START_RANGE client16y socket2181 140 31610-31624
START_RANGE client16z socket2182 150 31624-31639
START_RANGE client16aa socket2183 140 31639-31653
START_RANGE client16ab socket2184 150 31653-31668
START_RANGE client16o socket2185 140 31668-31682
START_RANGE client16p socket2186 150 31682-31697
START_RANGE client16q socket2187 140 31697-31711
START_RANGE client16r socket2188 150 31711-31726
START_RANGE client16s socket2189 140 31726-31740
START_RANGE client16t socket2190 150 31740-31755
START_RANGE client16u socket2191 140 31755-31769
START_RANGE client16v socket2192 150 31769-31784
START_RANGE client16w socket2193 140 31784-31798
START_RANGE client16x socket2194 150 31798-31813
START_RANGE client16y socket2195 140 31813-31827
START_RANGE client16z socket2196 150 31827-31842
START_RANGE client16aa socket2197 140 31842-31856
START_RANGE client16ab socket2198 150 31856-31871
START_RANGE client16o socket2199 140 31871-31885
START_RANGE client16p socket2200 150 31885-31900
START_RANGE client16q socket2201 140 31900-31914
START_RANGE client16r socket2202 150 31914-31929
START_RANGE client16s socket2203 140 31929-31943
START_RANGE client16t socket2204 150 31943-31958
START_RANGE client16u socket2205 140 31958-31972
START_RANGE client16v socket2206 150 31972-31987
START_RANGE client16w socket2207 140 31987-32001
START_RANGE client16x socket2208 150 32001-32016
START_RANGE client16y socket2209 140 32016-32030
```

```
START_RANGE client16z socket2210 150 32030-32045
START_RANGE client16aa socket2211 140 32045-32059
START_RANGE client16ab socket2212 150 32059-32074
START_RANGE client16o socket2213 140 32074-32088
START_RANGE client16p socket2214 150 32088-32103
START_RANGE client16q socket2215 140 32103-32117
START_RANGE client16r socket2216 150 32117-32132
START_RANGE client16s socket2217 140 32132-32146
START_RANGE client16t socket2218 150 32146-32161
START_RANGE client16u socket2219 140 32161-32175
START_RANGE client16v socket2220 150 32175-32190
START_RANGE client16w socket2221 140 32190-32204
START_RANGE client16x socket2222 150 32204-32219
START_RANGE client16y socket2223 140 32219-32233
START_RANGE client16z socket2224 150 32233-32248
START_RANGE client16aa socket2225 140 32248-32262
START_RANGE client16ab socket2226 150 32262-32277
START_RANGE client16o socket2227 140 32277-32291
START_RANGE client16p socket2228 150 32291-32306
START_RANGE client16q socket2229 140 32306-32320
START_RANGE client16r socket2230 150 32320-32335
START_RANGE client16s socket2231 140 32335-32349
START_RANGE client16t socket2232 150 32349-32364
START_RANGE client16u socket2233 140 32364-32378
START_RANGE client16v socket2234 150 32378-32393
START_RANGE client16w socket2235 140 32393-32407
START_RANGE client16x socket2236 150 32407-32422
START_RANGE client16y socket2237 140 32422-32436
START_RANGE client16z socket2238 150 32436-32451
START_RANGE client16aa socket2239 140 32451-32465
START_RANGE client16ab socket2240 150 32465-32480

START_RANGE client17a socket2241 140 32480-32494
START_RANGE client17b socket2242 150 32494-32509
START_RANGE client17c socket2243 140 32509-32523
START_RANGE client17d socket2244 150 32523-32538
START_RANGE client17e socket2245 140 32538-32552
START_RANGE client17f socket2246 150 32552-32567
START_RANGE client17g socket2247 140 32567-32581
START_RANGE client17h socket2248 150 32581-32596
START_RANGE client17i socket2249 140 32596-32610
START_RANGE client17j socket2250 150 32610-32625
START_RANGE client17k socket2251 140 32625-32639
START_RANGE client17l socket2252 150 32639-32654
START_RANGE client17m socket2253 140 32654-32668
START_RANGE client17n socket2254 150 32668-32683
START_RANGE client17a socket2255 140 32683-32697
START_RANGE client17b socket2256 150 32697-32712
START_RANGE client17c socket2257 140 32712-32726
START_RANGE client17d socket2258 150 32726-32741
START_RANGE client17e socket2259 140 32741-32755
START_RANGE client17f socket2260 150 32755-32770
START_RANGE client17g socket2261 140 32770-32784
START_RANGE client17h socket2262 150 32784-32799
START_RANGE client17i socket2263 140 32799-32813
START_RANGE client17j socket2264 150 32813-32828
START_RANGE client17k socket2265 140 32828-32842
START_RANGE client17l socket2266 150 32842-32857
START_RANGE client17m socket2267 140 32857-32871
START_RANGE client17n socket2268 150 32871-32886
START_RANGE client17a socket2269 140 32886-32900
START_RANGE client17b socket2270 150 32900-32915
START_RANGE client17c socket2271 140 32915-32929
START_RANGE client17d socket2272 150 32929-32944
START_RANGE client17e socket2273 140 32944-32958
START_RANGE client17f socket2274 150 32958-32973
START_RANGE client17g socket2275 140 32973-32987
START_RANGE client17h socket2276 150 32987-33002
START_RANGE client17i socket2277 140 33002-33016
START_RANGE client17j socket2278 150 33016-33031
START_RANGE client17k socket2279 140 33031-33045
START_RANGE client17l socket2280 150 33045-33060
START_RANGE client17m socket2281 140 33060-33074
START_RANGE client17n socket2282 150 33074-33089
START_RANGE client17a socket2283 140 33089-33103
START_RANGE client17b socket2284 150 33103-33118
START_RANGE client17c socket2285 140 33118-33132
START_RANGE client17d socket2286 150 33132-33147
START_RANGE client17e socket2287 140 33147-33161
START_RANGE client17f socket2288 150 33161-33176
START_RANGE client17g socket2289 140 33176-33190
START_RANGE client17h socket2290 150 33190-33205
START_RANGE client17i socket2291 140 33205-33219
START_RANGE client17j socket2292 150 33219-33234
START_RANGE client17k socket2293 140 33234-33248
START_RANGE client17l socket2294 150 33248-33263
START_RANGE client17m socket2295 140 33263-33277
START_RANGE client17n socket2296 150 33277-33292
START_RANGE client17a socket2297 140 33292-33306
START_RANGE client17b socket2298 150 33306-33321
START_RANGE client17c socket2299 140 33321-33335
START_RANGE client17d socket2300 150 33335-33350
START_RANGE client17e socket2301 140 33350-33364
START_RANGE client17f socket2302 150 33364-33379
START_RANGE client17g socket2303 140 33379-33393
START_RANGE client17h socket2304 150 33393-33408
START_RANGE client17i socket2305 140 33408-33422
START_RANGE client17j socket2306 150 33422-33437
START_RANGE client17k socket2307 140 33437-33451
START_RANGE client17l socket2308 150 33451-33466
START_RANGE client17m socket2309 140 33466-33480
START_RANGE client17n socket2310 150 33480-33495

START_RANGE client17o socket2311 140 33495-33509
START_RANGE client17p socket2312 150 33509-33524
START_RANGE client17q socket2313 140 33524-33538
START_RANGE client17r socket2314 150 33538-33553
START_RANGE client17s socket2315 140 33553-33567
START_RANGE client17t socket2316 150 33567-33582
START_RANGE client17u socket2317 140 33582-33596
START_RANGE client17v socket2318 150 33596-33611
START_RANGE client17w socket2319 140 33611-33625
START_RANGE client17x socket2320 150 33625-33640
START_RANGE client17y socket2321 140 33640-33654
START_RANGE client17z socket2322 150 33654-33669
START_RANGE client17aa socket2323 140 33669-33683
START_RANGE client17ab socket2324 150 33683-33698
START_RANGE client17o socket2325 140 33698-33712
START_RANGE client17p socket2326 150 33712-33727
START_RANGE client17q socket2327 140 33727-33741
START_RANGE client17r socket2328 150 33741-33756
START_RANGE client17s socket2329 140 33756-33770
START_RANGE client17t socket2330 150 33770-33785
START_RANGE client17u socket2331 140 33785-33799
START_RANGE client17v socket2332 150 33799-33814
START_RANGE client17w socket2333 140 33814-33828
START_RANGE client17x socket2334 150 33828-33843
START_RANGE client17y socket2335 140 33843-33857
START_RANGE client17z socket2336 150 33857-33872
START_RANGE client17aa socket2337 140 33872-33886
START_RANGE client17ab socket2338 150 33886-33901
START_RANGE client17o socket2339 140 33901-33915
START_RANGE client17p socket2340 150 33915-33930
START_RANGE client17q socket2341 140 33930-33944
START_RANGE client17r socket2342 150 33944-33959
START_RANGE client17s socket2343 140 33959-33973
START_RANGE client17t socket2344 150 33973-33988
START_RANGE client17u socket2345 140 33988-34002
START_RANGE client17v socket2346 150 34002-34017
START_RANGE client17w socket2347 140 34017-34031
START_RANGE client17x socket2348 150 34031-34046
START_RANGE client17y socket2349 140 34046-34060
START_RANGE client17z socket2350 150 34060-34075
START_RANGE client17aa socket2351 140 34075-34089
START_RANGE client17ab socket2352 150 34089-34104
START_RANGE client17o socket2353 140 34104-34118
START_RANGE client17p socket2354 150 34118-34133
START_RANGE client17q socket2355 140 34133-34147
START_RANGE client17r socket2356 150 34147-34162
START_RANGE client17s socket2357 140 34162-34176
START_RANGE client17t socket2358 150 34176-34191
START_RANGE client17u socket2359 140 34191-34205
START_RANGE client17v socket2360 150 34205-34220
START_RANGE client17w socket2361 140 34220-34234
START_RANGE client17x socket2362 150 34234-34249
START_RANGE client17y socket2363 140 34249-34263
START_RANGE client17z socket2364 150 34263-34278
START_RANGE client17aa socket2365 140 34278-34292
START_RANGE client17ab socket2366 150 34292-34307
START_RANGE client17o socket2367 140 34307-34321
START_RANGE client17p socket2368 150 34321-34336
START_RANGE client17q socket2369 140 34336-34350
START_RANGE client17r socket2370 150 34350-34365
START_RANGE client17s socket2371 140 34365-34379
START_RANGE client17t socket2372 150 34379-34394
START_RANGE client17u socket2373 140 34394-34408
START_RANGE client17v socket2374 150 34408-34423
START_RANGE client17w socket2375 140 34423-34437
START_RANGE client17x socket2376 150 34437-34452
START_RANGE client17y socket2377 140 34452-34466
START_RANGE client17z socket2378 150 34466-34481
START_RANGE client17aa socket2379 140 34481-34495
START_RANGE client17ab socket2380 150 34495-34510

START_RANGE client18a socket2381 140 34510-34524
START_RANGE client18b socket2382 150 34524-34539
START_RANGE client18c socket2383 140 34539-34553
START_RANGE client18d socket2384 150 34553-34568
START_RANGE client18e socket2385 140 34568-34582
START_RANGE client18f socket2386 150 34582-34597
START_RANGE client18g socket2387 140 34597-34611
START_RANGE client18h socket2388 150 34611-34626
START_RANGE client18i socket2389 140 34626-34640
START_RANGE client18j socket2390 150 34640-34655
START_RANGE client18k socket2391 140 34655-34669
START_RANGE client18l socket2392 150 34669-34684
START_RANGE client18m socket2393 140 34684-34698
START_RANGE client18n socket2394 150 34698-34713
START_RANGE client18a socket2395 140 34713-34727
START_RANGE client18b socket2396 150 34727-34742
START_RANGE client18c socket2397 140 34742-34756
START_RANGE client18d socket2398 150 34756-34771
START_RANGE client18e socket2399 140 34771-34785
START_RANGE client18f socket2400 150 34785-34800
START_RANGE client18g socket2401 140 34800-34814
START_RANGE client18h socket2402 150 34814-34829
START_RANGE client18i socket2403 140 34829-34843
START_RANGE client18j socket2404 150 34843-34858
```

```
START_RANGE client18k socket2405 140 34858-34872
START_RANGE client18l socket2406 150 34872-34887
START_RANGE client18m socket2407 140 34887-34901
START_RANGE client18n socket2408 150 34901-34916
START_RANGE client18a socket2409 140 34916-34930
START_RANGE client18b socket2410 150 34930-34945
START_RANGE client18c socket2411 140 34945-34959
START_RANGE client18d socket2412 150 34959-34974
START_RANGE client18e socket2413 140 34974-34988
START_RANGE client18f socket2414 150 34988-35003
START_RANGE client18g socket2415 140 35003-35017
START_RANGE client18h socket2416 150 35017-35032
START_RANGE client18i socket2417 140 35032-35046
START_RANGE client18j socket2418 150 35046-35061
START_RANGE client18k socket2419 140 35061-35075
START_RANGE client18l socket2420 150 35075-35090
START_RANGE client18m socket2421 140 35090-35104
START_RANGE client18n socket2422 150 35104-35119
START_RANGE client18a socket2423 140 35119-35133
START_RANGE client18b socket2424 150 35133-35148
START_RANGE client18c socket2425 140 35148-35162
START_RANGE client18d socket2426 150 35162-35177
START_RANGE client18e socket2427 140 35177-35191
START_RANGE client18f socket2428 150 35191-35206
START_RANGE client18g socket2429 140 35206-35220
START_RANGE client18h socket2430 150 35220-35235
START_RANGE client18i socket2431 140 35235-35249
START_RANGE client18j socket2432 150 35249-35264
START_RANGE client18k socket2433 140 35264-35278
START_RANGE client18l socket2434 150 35278-35293
START_RANGE client18m socket2435 140 35293-35307
START_RANGE client18n socket2436 150 35307-35322
START_RANGE client18a socket2437 140 35322-35336
START_RANGE client18b socket2438 150 35336-35351
START_RANGE client18c socket2439 140 35351-35365
START_RANGE client18d socket2440 150 35365-35380
START_RANGE client18e socket2441 140 35380-35394
START_RANGE client18f socket2442 150 35394-35409
START_RANGE client18g socket2443 140 35409-35423
START_RANGE client18h socket2444 150 35423-35438
START_RANGE client18i socket2445 140 35438-35452
START_RANGE client18j socket2446 150 35452-35467
START_RANGE client18k socket2447 140 35467-35481
START_RANGE client18l socket2448 150 35481-35496
START_RANGE client18m socket2449 140 35496-35510
START_RANGE client18n socket2450 150 35510-35525

START_RANGE client18o socket2451 140 35525-35539
START_RANGE client18p socket2452 150 35539-35554
START_RANGE client18q socket2453 140 35554-35568
START_RANGE client18r socket2454 150 35568-35583
START_RANGE client18s socket2455 140 35583-35597
START_RANGE client18t socket2456 150 35597-35612
START_RANGE client18u socket2457 140 35612-35626
START_RANGE client18v socket2458 150 35626-35641
START_RANGE client18w socket2459 140 35641-35655
START_RANGE client18x socket2460 150 35655-35670
START_RANGE client18y socket2461 140 35670-35684
START_RANGE client18z socket2462 150 35684-35699
START_RANGE client18aa socket2463 140 35699-35713
START_RANGE client18ab socket2464 150 35713-35728
START_RANGE client18o socket2465 140 35728-35742
START_RANGE client18p socket2466 150 35742-35757
START_RANGE client18q socket2467 140 35757-35771
START_RANGE client18r socket2468 150 35771-35786
START_RANGE client18s socket2469 140 35786-35800
START_RANGE client18t socket2470 150 35800-35815
START_RANGE client18u socket2471 140 35815-35829
START_RANGE client18v socket2472 150 35829-35844
START_RANGE client18w socket2473 140 35844-35858
START_RANGE client18x socket2474 150 35858-35873
START_RANGE client18y socket2475 140 35873-35887
START_RANGE client18z socket2476 150 35887-35902
START_RANGE client18aa socket2477 140 35902-35916
START_RANGE client18ab socket2478 150 35916-35931
START_RANGE client18o socket2479 140 35931-35945
START_RANGE client18p socket2480 150 35945-35960
START_RANGE client18q socket2481 140 35960-35974
START_RANGE client18r socket2482 150 35974-35989
START_RANGE client18s socket2483 140 35989-36003
START_RANGE client18t socket2484 150 36003-36018
START_RANGE client18u socket2485 140 36018-36032
START_RANGE client18v socket2486 150 36032-36047
START_RANGE client18w socket2487 140 36047-36061
START_RANGE client18x socket2488 150 36061-36076
START_RANGE client18y socket2489 140 36076-36090
START_RANGE client18z socket2490 150 36090-36105
START_RANGE client18aa socket2491 140 36105-36119
START_RANGE client18ab socket2492 150 36119-36134
START_RANGE client18o socket2493 140 36134-36148
START_RANGE client18p socket2494 150 36148-36163
START_RANGE client18q socket2495 140 36163-36177
START_RANGE client18r socket2496 150 36177-36192
START_RANGE client18s socket2497 140 36192-36206
START_RANGE client18t socket2498 150 36206-36221
START_RANGE client18u socket2499 140 36221-36235
START_RANGE client18v socket2500 150 36235-36250
START_RANGE client18w socket2501 140 36250-36264
START_RANGE client18x socket2502 150 36264-36279
START_RANGE client18y socket2503 140 36279-36293
START_RANGE client18z socket2504 150 36293-36308
START_RANGE client18aa socket2505 140 36308-36322
START_RANGE client18ab socket2506 150 36322-36337
START_RANGE client18o socket2507 140 36337-36351
START_RANGE client18p socket2508 150 36351-36366
START_RANGE client18q socket2509 140 36366-36380
START_RANGE client18r socket2510 150 36380-36395
START_RANGE client18s socket2511 140 36395-36409
START_RANGE client18t socket2512 150 36409-36424
START_RANGE client18u socket2513 140 36424-36438
START_RANGE client18v socket2514 150 36438-36453
START_RANGE client18w socket2515 140 36453-36467
START_RANGE client18x socket2516 150 36467-36482
START_RANGE client18y socket2517 140 36482-36496
START_RANGE client18z socket2518 150 36496-36511
START_RANGE client18aa socket2519 140 36511-36525
START_RANGE client18ab socket2520 150 36525-36540

#elif MASTER_NUM7
START_RANGE client19a socket2521 140 36540-36554
START_RANGE client19b socket2522 150 36554-36569
START_RANGE client19c socket2523 140 36569-36583
START_RANGE client19d socket2524 150 36583-36598
START_RANGE client19e socket2525 140 36598-36612
START_RANGE client19f socket2526 150 36612-36627
START_RANGE client19g socket2527 140 36627-36641
START_RANGE client19h socket2528 150 36641-36656
START_RANGE client19i socket2529 140 36656-36670
START_RANGE client19j socket2530 150 36670-36685
START_RANGE client19k socket2531 140 36685-36699
START_RANGE client19l socket2532 150 36699-36714
START_RANGE client19m socket2533 140 36714-36728
START_RANGE client19n socket2534 150 36728-36743
START_RANGE client19a socket2535 140 36743-36757
START_RANGE client19b socket2536 150 36757-36772
START_RANGE client19c socket2537 140 36772-36786
START_RANGE client19d socket2538 150 36786-36801
START_RANGE client19e socket2539 140 36801-36815
START_RANGE client19f socket2540 150 36815-36830
START_RANGE client19g socket2541 140 36830-36844
START_RANGE client19h socket2542 150 36844-36859
START_RANGE client19i socket2543 140 36859-36873
START_RANGE client19j socket2544 150 36873-36888
START_RANGE client19k socket2545 140 36888-36902
START_RANGE client19l socket2546 150 36902-36917
START_RANGE client19m socket2547 140 36917-36931
START_RANGE client19n socket2548 150 36931-36946
START_RANGE client19a socket2549 140 36946-36960
START_RANGE client19b socket2550 150 36960-36975
START_RANGE client19c socket2551 140 36975-36989
START_RANGE client19d socket2552 150 36989-37004
START_RANGE client19e socket2553 140 37004-37018
START_RANGE client19f socket2554 150 37018-37033
START_RANGE client19g socket2555 140 37033-37047
START_RANGE client19h socket2556 150 37047-37062
START_RANGE client19i socket2557 140 37062-37076
START_RANGE client19j socket2558 150 37076-37091
START_RANGE client19k socket2559 140 37091-37105
START_RANGE client19l socket2560 150 37105-37120
START_RANGE client19m socket2561 140 37120-37134
START_RANGE client19n socket2562 150 37134-37149
START_RANGE client19a socket2563 140 37149-37163
START_RANGE client19b socket2564 150 37163-37178
START_RANGE client19c socket2565 140 37178-37192
START_RANGE client19d socket2566 150 37192-37207
START_RANGE client19e socket2567 140 37207-37221
START_RANGE client19f socket2568 150 37221-37236
START_RANGE client19g socket2569 140 37236-37250
START_RANGE client19h socket2570 150 37250-37265
START_RANGE client19i socket2571 140 37265-37279
START_RANGE client19j socket2572 150 37279-37294
START_RANGE client19k socket2573 140 37294-37308
START_RANGE client19l socket2574 150 37308-37323
START_RANGE client19m socket2575 140 37323-37337
START_RANGE client19n socket2576 150 37337-37352
START_RANGE client19a socket2577 140 37352-37366
START_RANGE client19b socket2578 150 37366-37381
START_RANGE client19c socket2579 140 37381-37395
START_RANGE client19d socket2580 150 37395-37410
START_RANGE client19e socket2581 140 37410-37424
START_RANGE client19f socket2582 150 37424-37439
START_RANGE client19g socket2583 140 37439-37453
START_RANGE client19h socket2584 150 37453-37468
START_RANGE client19i socket2585 140 37468-37482
START_RANGE client19j socket2586 150 37482-37497
START_RANGE client19k socket2587 140 37497-37511
START_RANGE client19l socket2588 150 37511-37526
START_RANGE client19m socket2589 140 37526-37540
START_RANGE client19n socket2590 150 37540-37555

START_RANGE client19o socket2591 140 37555-37569
START_RANGE client19p socket2592 150 37569-37584
START_RANGE client19q socket2593 140 37584-37598
START_RANGE client19r socket2594 150 37598-37613
START_RANGE client19s socket2595 140 37613-37627
START_RANGE client19t socket2596 150 37627-37642
START_RANGE client19u socket2597 140 37642-37656
START_RANGE client19v socket2598 150 37656-37671
```

```
START_RANGE client19w socket2599 140 37671-37685
START_RANGE client19x socket2600 150 37685-37700
START_RANGE client19y socket2601 140 37700-37714
START_RANGE client19z socket2602 150 37714-37729
START_RANGE client19aa socket2603 140 37729-37743
START_RANGE client19ab socket2604 150 37743-37758
START_RANGE client19o socket2605 140 37758-37772
START_RANGE client19p socket2606 150 37772-37787
START_RANGE client19q socket2607 140 37787-37801
START_RANGE client19r socket2608 150 37801-37816
START_RANGE client19s socket2609 140 37816-37830
START_RANGE client19t socket2610 150 37830-37845
START_RANGE client19u socket2611 140 37845-37859
START_RANGE client19v socket2612 150 37859-37874
START_RANGE client19w socket2613 140 37874-37888
START_RANGE client19x socket2614 150 37888-37903
START_RANGE client19y socket2615 140 37903-37917
START_RANGE client19z socket2616 150 37917-37932
START_RANGE client19aa socket2617 140 37932-37946
START_RANGE client19ab socket2618 150 37946-37961
START_RANGE client19o socket2619 140 37961-37975
START_RANGE client19p socket2620 150 37975-37990
START_RANGE client19q socket2621 140 37990-38004
START_RANGE client19r socket2622 150 38004-38019
START_RANGE client19s socket2623 140 38019-38033
START_RANGE client19t socket2624 150 38033-38048
START_RANGE client19u socket2625 140 38048-38062
START_RANGE client19v socket2626 150 38062-38077
START_RANGE client19w socket2627 140 38077-38091
START_RANGE client19x socket2628 150 38091-38106
START_RANGE client19y socket2629 140 38106-38120
START_RANGE client19z socket2630 150 38120-38135
START_RANGE client19aa socket2631 140 38135-38149
START_RANGE client19ab socket2632 150 38149-38164
START_RANGE client19o socket2633 140 38164-38178
START_RANGE client19p socket2634 150 38178-38193
START_RANGE client19q socket2635 140 38193-38207
START_RANGE client19r socket2636 150 38207-38222
START_RANGE client19s socket2637 140 38222-38236
START_RANGE client19t socket2638 150 38236-38251
START_RANGE client19u socket2639 140 38251-38265
START_RANGE client19v socket2640 150 38265-38280
START_RANGE client19w socket2641 140 38280-38294
START_RANGE client19x socket2642 150 38294-38309
START_RANGE client19y socket2643 140 38309-38323
START_RANGE client19z socket2644 150 38323-38338
START_RANGE client19aa socket2645 140 38338-38352
START_RANGE client19ab socket2646 150 38352-38367
START_RANGE client19o socket2647 140 38367-38381
START_RANGE client19p socket2648 150 38381-38396
START_RANGE client19q socket2649 140 38396-38410
START_RANGE client19r socket2650 150 38410-38425
START_RANGE client19s socket2651 140 38425-38439
START_RANGE client19t socket2652 150 38439-38454
START_RANGE client19u socket2653 140 38454-38468
START_RANGE client19v socket2654 150 38468-38483
START_RANGE client19w socket2655 140 38483-38497
START_RANGE client19x socket2656 150 38497-38512
START_RANGE client19y socket2657 140 38512-38526
START_RANGE client19z socket2658 150 38526-38541
START_RANGE client19aa socket2659 140 38541-38555
START_RANGE client19ab socket2660 150 38555-38570

START_RANGE client20a socket2661 140 38570-38584
START_RANGE client20b socket2662 150 38584-38599
START_RANGE client20c socket2663 140 38599-38613
START_RANGE client20d socket2664 150 38613-38628
START_RANGE client20e socket2665 140 38628-38642
START_RANGE client20f socket2666 150 38642-38657
START_RANGE client20g socket2667 140 38657-38671
START_RANGE client20h socket2668 150 38671-38686
START_RANGE client20i socket2669 140 38686-38700
START_RANGE client20j socket2670 150 38700-38715
START_RANGE client20k socket2671 140 38715-38729
START_RANGE client20l socket2672 150 38729-38744
START_RANGE client20m socket2673 140 38744-38758
START_RANGE client20n socket2674 150 38758-38773
START_RANGE client20a socket2675 140 38773-38787
START_RANGE client20b socket2676 150 38787-38802
START_RANGE client20c socket2677 140 38802-38816
START_RANGE client20d socket2678 150 38816-38831
START_RANGE client20e socket2679 140 38831-38845
START_RANGE client20f socket2680 150 38845-38860
START_RANGE client20g socket2681 140 38860-38874
START_RANGE client20h socket2682 150 38874-38889
START_RANGE client20i socket2683 140 38889-38903
START_RANGE client20j socket2684 150 38903-38918
START_RANGE client20k socket2685 140 38918-38932
START_RANGE client20l socket2686 150 38932-38947
START_RANGE client20m socket2687 140 38947-38961
START_RANGE client20n socket2688 150 38961-38976
START_RANGE client20a socket2689 140 38976-38990
START_RANGE client20b socket2690 150 38990-39005
START_RANGE client20c socket2691 140 39005-39019
START_RANGE client20d socket2692 150 39019-39034
START_RANGE client20e socket2693 140 39034-39048
START_RANGE client20f socket2694 150 39048-39063
START_RANGE client20g socket2695 140 39063-39077
START_RANGE client20h socket2696 150 39077-39092
START_RANGE client20i socket2697 140 39092-39106
START_RANGE client20j socket2698 150 39106-39121
START_RANGE client20k socket2699 140 39121-39135
START_RANGE client20l socket2700 150 39135-39150
START_RANGE client20m socket2701 140 39150-39164
START_RANGE client20n socket2702 150 39164-39179
START_RANGE client20a socket2703 140 39179-39193
START_RANGE client20b socket2704 150 39193-39208
START_RANGE client20c socket2705 140 39208-39222
START_RANGE client20d socket2706 150 39222-39237
START_RANGE client20e socket2707 140 39237-39251
START_RANGE client20f socket2708 150 39251-39266
START_RANGE client20g socket2709 140 39266-39280
START_RANGE client20h socket2710 150 39280-39295
START_RANGE client20i socket2711 140 39295-39309
START_RANGE client20j socket2712 150 39309-39324
START_RANGE client20k socket2713 140 39324-39338
START_RANGE client20l socket2714 150 39338-39353
START_RANGE client20m socket2715 140 39353-39367
START_RANGE client20n socket2716 150 39367-39382
START_RANGE client20a socket2717 140 39382-39396
START_RANGE client20b socket2718 150 39396-39411
START_RANGE client20c socket2719 140 39411-39425
START_RANGE client20d socket2720 150 39425-39440
START_RANGE client20e socket2721 140 39440-39454
START_RANGE client20f socket2722 150 39454-39469
START_RANGE client20g socket2723 140 39469-39483
START_RANGE client20h socket2724 150 39483-39498
START_RANGE client20i socket2725 140 39498-39512
START_RANGE client20j socket2726 150 39512-39527
START_RANGE client20k socket2727 140 39527-39541
START_RANGE client20l socket2728 150 39541-39556
START_RANGE client20m socket2729 140 39556-39570
START_RANGE client20n socket2730 150 39570-39585

START_RANGE client20o socket2731 140 39585-39599
START_RANGE client20p socket2732 150 39599-39614
START_RANGE client20q socket2733 140 39614-39628
START_RANGE client20r socket2734 150 39628-39643
START_RANGE client20s socket2735 140 39643-39657
START_RANGE client20t socket2736 150 39657-39672
START_RANGE client20u socket2737 140 39672-39686
START_RANGE client20v socket2738 150 39686-39701
START_RANGE client20w socket2739 140 39701-39715
START_RANGE client20x socket2740 150 39715-39730
START_RANGE client20y socket2741 140 39730-39744
START_RANGE client20z socket2742 150 39744-39759
START_RANGE client20aa socket2743 140 39759-39773
START_RANGE client20ab socket2744 150 39773-39788
START_RANGE client20o socket2745 140 39788-39802
START_RANGE client20p socket2746 150 39802-39817
START_RANGE client20q socket2747 140 39817-39831
START_RANGE client20r socket2748 150 39831-39846
START_RANGE client20s socket2749 140 39846-39860
START_RANGE client20t socket2750 150 39860-39875
START_RANGE client20u socket2751 140 39875-39889
START_RANGE client20v socket2752 150 39889-39904
START_RANGE client20w socket2753 140 39904-39918
START_RANGE client20x socket2754 150 39918-39933
START_RANGE client20y socket2755 140 39933-39947
START_RANGE client20z socket2756 150 39947-39962
START_RANGE client20aa socket2757 140 39962-39976
START_RANGE client20ab socket2758 150 39976-39991
START_RANGE client20o socket2759 140 39991-40005
START_RANGE client20p socket2760 150 40005-40020
START_RANGE client20q socket2761 140 40020-40034
START_RANGE client20r socket2762 150 40034-40049
START_RANGE client20s socket2763 140 40049-40063
START_RANGE client20t socket2764 150 40063-40078
START_RANGE client20u socket2765 140 40078-40092
START_RANGE client20v socket2766 150 40092-40107
START_RANGE client20w socket2767 140 40107-40121
START_RANGE client20x socket2768 150 40121-40136
START_RANGE client20y socket2769 140 40136-40150
START_RANGE client20z socket2770 150 40150-40165
START_RANGE client20aa socket2771 140 40165-40179
START_RANGE client20ab socket2772 150 40179-40194
START_RANGE client20o socket2773 140 40194-40208
START_RANGE client20p socket2774 150 40208-40223
START_RANGE client20q socket2775 140 40223-40237
START_RANGE client20r socket2776 150 40237-40252
START_RANGE client20s socket2777 140 40252-40266
START_RANGE client20t socket2778 150 40266-40281
START_RANGE client20u socket2779 140 40281-40295
START_RANGE client20v socket2780 150 40295-40310
START_RANGE client20w socket2781 140 40310-40324
START_RANGE client20x socket2782 150 40324-40339
START_RANGE client20y socket2783 140 40339-40353
START_RANGE client20z socket2784 150 40353-40368
START_RANGE client20aa socket2785 140 40368-40382
START_RANGE client20ab socket2786 150 40382-40397
START_RANGE client20o socket2787 140 40397-40411
START_RANGE client20p socket2788 150 40411-40426
START_RANGE client20q socket2789 140 40426-40440
START_RANGE client20r socket2790 150 40440-40455
START_RANGE client20s socket2791 140 40455-40469
START_RANGE client20t socket2792 150 40469-40484
START_RANGE client20u socket2793 140 40484-40498
START_RANGE client20v socket2794 150 40498-40513
```

```
START_RANGE client20w socket2795 140 40513-40527
START_RANGE client20x socket2796 150 40527-40542
START_RANGE client20y socket2797 140 40542-40556
START_RANGE client20z socket2798 150 40556-40571
START_RANGE client20aa socket2799 140 40571-40585
START_RANGE client20ab socket2800 150 40585-40600

START_RANGE client21a socket2801 140 40600-40614
START_RANGE client21b socket2802 150 40614-40629
START_RANGE client21c socket2803 140 40629-40643
START_RANGE client21d socket2804 150 40643-40658
START_RANGE client21e socket2805 140 40658-40672
START_RANGE client21f socket2806 150 40672-40687
START_RANGE client21g socket2807 140 40687-40701
START_RANGE client21h socket2808 150 40701-40716
START_RANGE client21i socket2809 140 40716-40730
START_RANGE client21j socket2810 150 40730-40745
START_RANGE client21k socket2811 140 40745-40759
START_RANGE client21l socket2812 150 40759-40774
START_RANGE client21m socket2813 140 40774-40788
START_RANGE client21n socket2814 150 40788-40803
START_RANGE client21a socket2815 140 40803-40817
START_RANGE client21b socket2816 150 40817-40832
START_RANGE client21c socket2817 140 40832-40846
START_RANGE client21d socket2818 150 40846-40861
START_RANGE client21e socket2819 140 40861-40875
START_RANGE client21f socket2820 150 40875-40890
START_RANGE client21g socket2821 140 40890-40904
START_RANGE client21h socket2822 150 40904-40919
START_RANGE client21i socket2823 140 40919-40933
START_RANGE client21j socket2824 150 40933-40948
START_RANGE client21k socket2825 140 40948-40962
START_RANGE client21l socket2826 150 40962-40977
START_RANGE client21m socket2827 140 40977-40991
START_RANGE client21n socket2828 150 40991-41006
START_RANGE client21a socket2829 140 41006-41020
START_RANGE client21b socket2830 150 41020-41035
START_RANGE client21c socket2831 140 41035-41049
START_RANGE client21d socket2832 150 41049-41064
START_RANGE client21e socket2833 140 41064-41078
START_RANGE client21f socket2834 150 41078-41093
START_RANGE client21g socket2835 140 41093-41107
START_RANGE client21h socket2836 150 41107-41122
START_RANGE client21i socket2837 140 41122-41136
START_RANGE client21j socket2838 150 41136-41151
START_RANGE client21k socket2839 140 41151-41165
START_RANGE client21l socket2840 150 41165-41180
START_RANGE client21m socket2841 140 41180-41194
START_RANGE client21n socket2842 150 41194-41209
START_RANGE client21a socket2843 140 41209-41223
START_RANGE client21b socket2844 150 41223-41238
START_RANGE client21c socket2845 140 41238-41252
START_RANGE client21d socket2846 150 41252-41267
START_RANGE client21e socket2847 140 41267-41281
START_RANGE client21f socket2848 150 41281-41296
START_RANGE client21g socket2849 140 41296-41310
START_RANGE client21h socket2850 150 41310-41325
START_RANGE client21i socket2851 140 41325-41339
START_RANGE client21j socket2852 150 41339-41354
START_RANGE client21k socket2853 140 41354-41368
START_RANGE client21l socket2854 150 41368-41383
START_RANGE client21m socket2855 140 41383-41397
START_RANGE client21n socket2856 150 41397-41412
START_RANGE client21a socket2857 140 41412-41426
START_RANGE client21b socket2858 150 41426-41441
START_RANGE client21c socket2859 140 41441-41455
START_RANGE client21d socket2860 150 41455-41470
START_RANGE client21e socket2861 140 41470-41484
START_RANGE client21f socket2862 150 41484-41499
START_RANGE client21g socket2863 140 41499-41513
START_RANGE client21h socket2864 150 41513-41528
START_RANGE client21i socket2865 140 41528-41542
START_RANGE client21j socket2866 150 41542-41557
START_RANGE client21k socket2867 140 41557-41571
START_RANGE client21l socket2868 150 41571-41586
START_RANGE client21m socket2869 140 41586-41600
START_RANGE client21n socket2870 150 41600-41615

START_RANGE client21o socket2871 140 41615-41629
START_RANGE client21p socket2872 150 41629-41644
START_RANGE client21q socket2873 140 41644-41658
START_RANGE client21r socket2874 150 41658-41673
START_RANGE client21s socket2875 140 41673-41687
START_RANGE client21t socket2876 150 41687-41702
START_RANGE client21u socket2877 140 41702-41716
START_RANGE client21v socket2878 150 41716-41731
START_RANGE client21w socket2879 140 41731-41745
START_RANGE client21x socket2880 150 41745-41760
START_RANGE client21y socket2881 140 41760-41774
START_RANGE client21z socket2882 150 41774-41789
START_RANGE client21aa socket2883 140 41789-41803
START_RANGE client21ab socket2884 150 41803-41818
START_RANGE client21o socket2885 140 41818-41832
START_RANGE client21p socket2886 150 41832-41847
START_RANGE client21q socket2887 140 41847-41861
START_RANGE client21r socket2888 150 41861-41876
START_RANGE client21s socket2889 140 41876-41890
START_RANGE client21t socket2890 150 41890-41905
START_RANGE client21u socket2891 140 41905-41919
START_RANGE client21v socket2892 150 41919-41934
START_RANGE client21w socket2893 140 41934-41948
START_RANGE client21x socket2894 150 41948-41963
START_RANGE client21y socket2895 140 41963-41977
START_RANGE client21z socket2896 150 41977-41992
START_RANGE client21aa socket2897 140 41992-42006
START_RANGE client21ab socket2898 150 42006-42021
START_RANGE client21o socket2899 140 42021-42035
START_RANGE client21p socket2900 150 42035-42050
START_RANGE client21q socket2901 140 42050-42064
START_RANGE client21r socket2902 150 42064-42079
START_RANGE client21s socket2903 140 42079-42093
START_RANGE client21t socket2904 150 42093-42108
START_RANGE client21u socket2905 140 42108-42122
START_RANGE client21v socket2906 150 42122-42137
START_RANGE client21w socket2907 140 42137-42151
START_RANGE client21x socket2908 150 42151-42166
START_RANGE client21y socket2909 140 42166-42180
START_RANGE client21z socket2910 150 42180-42195
START_RANGE client21aa socket2911 140 42195-42209
START_RANGE client21ab socket2912 150 42209-42224
START_RANGE client21o socket2913 140 42224-42238
START_RANGE client21p socket2914 150 42238-42253
START_RANGE client21q socket2915 140 42253-42267
START_RANGE client21r socket2916 150 42267-42282
START_RANGE client21s socket2917 140 42282-42296
START_RANGE client21t socket2918 150 42296-42311
START_RANGE client21u socket2919 140 42311-42325
START_RANGE client21v socket2920 150 42325-42340
START_RANGE client21w socket2921 140 42340-42354
START_RANGE client21x socket2922 150 42354-42369
START_RANGE client21y socket2923 140 42369-42383
START_RANGE client21z socket2924 150 42383-42398
START_RANGE client21aa socket2925 140 42398-42412
START_RANGE client21ab socket2926 150 42412-42427
START_RANGE client21o socket2927 140 42427-42441
START_RANGE client21p socket2928 150 42441-42456
START_RANGE client21q socket2929 140 42456-42470
START_RANGE client21r socket2930 150 42470-42485
START_RANGE client21s socket2931 140 42485-42499
START_RANGE client21t socket2932 150 42499-42514
START_RANGE client21u socket2933 140 42514-42528
START_RANGE client21v socket2934 150 42528-42543
START_RANGE client21w socket2935 140 42543-42557
START_RANGE client21x socket2936 150 42557-42572
START_RANGE client21y socket2937 140 42572-42586
START_RANGE client21z socket2938 150 42586-42601
START_RANGE client21aa socket2939 140 42601-42615
START_RANGE client21ab socket2940 150 42615-42630

#elif MASTER_NUM8
START_RANGE client22a socket2941 140 42630-42644
START_RANGE client22b socket2942 150 42644-42659
START_RANGE client22c socket2943 140 42659-42673
START_RANGE client22d socket2944 150 42673-42688
START_RANGE client22e socket2945 140 42688-42702
START_RANGE client22f socket2946 150 42702-42717
START_RANGE client22g socket2947 140 42717-42731
START_RANGE client22h socket2948 150 42731-42746
START_RANGE client22i socket2949 140 42746-42760
START_RANGE client22j socket2950 150 42760-42775
START_RANGE client22k socket2951 140 42775-42789
START_RANGE client22l socket2952 150 42789-42804
START_RANGE client22m socket2953 140 42804-42818
START_RANGE client22n socket2954 150 42818-42833
START_RANGE client22a socket2955 140 42833-42847
START_RANGE client22b socket2956 150 42847-42862
START_RANGE client22c socket2957 140 42862-42876
START_RANGE client22d socket2958 150 42876-42891
START_RANGE client22e socket2959 140 42891-42905
START_RANGE client22f socket2960 150 42905-42920
START_RANGE client22g socket2961 140 42920-42934
START_RANGE client22h socket2962 150 42934-42949
START_RANGE client22i socket2963 140 42949-42963
START_RANGE client22j socket2964 150 42963-42978
START_RANGE client22k socket2965 140 42978-42992
START_RANGE client22l socket2966 150 42992-43007
START_RANGE client22m socket2967 140 43007-43021
START_RANGE client22n socket2968 150 43021-43036
START_RANGE client22a socket2969 140 43036-43050
START_RANGE client22b socket2970 150 43050-43065
START_RANGE client22c socket2971 140 43065-43079
START_RANGE client22d socket2972 150 43079-43094
START_RANGE client22e socket2973 140 43094-43108
START_RANGE client22f socket2974 150 43108-43123
START_RANGE client22g socket2975 140 43123-43137
START_RANGE client22h socket2976 150 43137-43152
START_RANGE client22i socket2977 140 43152-43166
START_RANGE client22j socket2978 150 43166-43181
START_RANGE client22k socket2979 140 43181-43195
START_RANGE client22l socket2980 150 43195-43210
START_RANGE client22m socket2981 140 43210-43224
START_RANGE client22n socket2982 150 43224-43239
START_RANGE client22a socket2983 140 43239-43253
START_RANGE client22b socket2984 150 43253-43268
START_RANGE client22c socket2985 140 43268-43282
START_RANGE client22d socket2986 150 43282-43297
START_RANGE client22e socket2987 140 43297-43311
START_RANGE client22f socket2988 150 43311-43326
```

```
START_RANGE client22g socket2989 140 43326-43340
START_RANGE client22h socket2990 150 43340-43355
START_RANGE client22i socket2991 140 43355-43369
START_RANGE client22j socket2992 150 43369-43384
START_RANGE client22k socket2993 140 43384-43398
START_RANGE client22l socket2994 150 43398-43413
START_RANGE client22m socket2995 140 43413-43427
START_RANGE client22n socket2996 150 43427-43442
START_RANGE client22a socket2997 140 43442-43456
START_RANGE client22b socket2998 150 43456-43471
START_RANGE client22c socket2999 140 43471-43485
START_RANGE client22d socket3000 150 43485-43500
START_RANGE client22e socket3001 140 43500-43514
START_RANGE client22f socket3002 150 43514-43529
START_RANGE client22g socket3003 140 43529-43543
START_RANGE client22h socket3004 150 43543-43558
START_RANGE client22i socket3005 140 43558-43572
START_RANGE client22j socket3006 150 43572-43587
START_RANGE client22k socket3007 140 43587-43601
START_RANGE client22l socket3008 150 43601-43616
START_RANGE client22m socket3009 140 43616-43630
START_RANGE client22n socket3010 150 43630-43645

START_RANGE client22o socket3011 140 43645-43659
START_RANGE client22p socket3012 150 43659-43674
START_RANGE client22q socket3013 140 43674-43688
START_RANGE client22r socket3014 150 43688-43703
START_RANGE client22s socket3015 140 43703-43717
START_RANGE client22t socket3016 150 43717-43732
START_RANGE client22u socket3017 140 43732-43746
START_RANGE client22v socket3018 150 43746-43761
START_RANGE client22w socket3019 140 43761-43775
START_RANGE client22x socket3020 150 43775-43790
START_RANGE client22y socket3021 140 43790-43804
START_RANGE client22z socket3022 150 43804-43819
START_RANGE client22aa socket3023 140 43819-43833
START_RANGE client22ab socket3024 150 43833-43848
START_RANGE client22o socket3025 140 43848-43862
START_RANGE client22p socket3026 150 43862-43877
START_RANGE client22q socket3027 140 43877-43891
START_RANGE client22r socket3028 150 43891-43906
START_RANGE client22s socket3029 140 43906-43920
START_RANGE client22t socket3030 150 43920-43935
START_RANGE client22u socket3031 140 43935-43949
START_RANGE client22v socket3032 150 43949-43964
START_RANGE client22w socket3033 140 43964-43978
START_RANGE client22x socket3034 150 43978-43993
START_RANGE client22y socket3035 140 43993-44007
START_RANGE client22z socket3036 150 44007-44022
START_RANGE client22aa socket3037 140 44022-44036
START_RANGE client22ab socket3038 150 44036-44051
START_RANGE client22o socket3039 140 44051-44065
START_RANGE client22p socket3040 150 44065-44080
START_RANGE client22q socket3041 140 44080-44094
START_RANGE client22r socket3042 150 44094-44109
START_RANGE client22s socket3043 140 44109-44123
START_RANGE client22t socket3044 150 44123-44138
START_RANGE client22u socket3045 140 44138-44152
START_RANGE client22v socket3046 150 44152-44167
START_RANGE client22w socket3047 140 44167-44181
START_RANGE client22x socket3048 150 44181-44196
START_RANGE client22y socket3049 140 44196-44210
START_RANGE client22z socket3050 150 44210-44225
START_RANGE client22aa socket3051 140 44225-44239
START_RANGE client22ab socket3052 150 44239-44254
START_RANGE client22o socket3053 140 44254-44268
START_RANGE client22p socket3054 150 44268-44283
START_RANGE client22q socket3055 140 44283-44297
START_RANGE client22r socket3056 150 44297-44312
START_RANGE client22s socket3057 140 44312-44326
START_RANGE client22t socket3058 150 44326-44341
START_RANGE client22u socket3059 140 44341-44355
START_RANGE client22v socket3060 150 44355-44370
START_RANGE client22w socket3061 140 44370-44384
START_RANGE client22x socket3062 150 44384-44399
START_RANGE client22y socket3063 140 44399-44413
START_RANGE client22z socket3064 150 44413-44428
START_RANGE client22aa socket3065 140 44428-44442
START_RANGE client22ab socket3066 150 44442-44457
START_RANGE client22o socket3067 140 44457-44471
START_RANGE client22p socket3068 150 44471-44486
START_RANGE client22q socket3069 140 44486-44500
START_RANGE client22r socket3070 150 44500-44515
START_RANGE client22s socket3071 140 44515-44529
START_RANGE client22t socket3072 150 44529-44544
START_RANGE client22u socket3073 140 44544-44558
START_RANGE client22v socket3074 150 44558-44573
START_RANGE client22w socket3075 140 44573-44587
START_RANGE client22x socket3076 150 44587-44602
START_RANGE client22y socket3077 140 44602-44616
START_RANGE client22z socket3078 150 44616-44631
START_RANGE client22aa socket3079 140 44631-44645
START_RANGE client22ab socket3080 150 44645-44660

START_RANGE client23a socket3081 140 44660-44674
START_RANGE client23b socket3082 150 44674-44689
START_RANGE client23c socket3083 140 44689-44703
START_RANGE client23d socket3084 150 44703-44718
START_RANGE client23e socket3085 140 44718-44732
START_RANGE client23f socket3086 150 44732-44747
START_RANGE client23g socket3087 140 44747-44761
START_RANGE client23h socket3088 150 44761-44776
START_RANGE client23i socket3089 140 44776-44790
START_RANGE client23j socket3090 150 44790-44805
START_RANGE client23k socket3091 140 44805-44819
START_RANGE client23l socket3092 150 44819-44834
START_RANGE client23m socket3093 140 44834-44848
START_RANGE client23n socket3094 150 44848-44863
START_RANGE client23a socket3095 140 44863-44877
START_RANGE client23b socket3096 150 44877-44892
START_RANGE client23c socket3097 140 44892-44906
START_RANGE client23d socket3098 150 44906-44921
START_RANGE client23e socket3099 140 44921-44935
START_RANGE client23f socket3100 150 44935-44950
START_RANGE client23g socket3101 140 44950-44964
START_RANGE client23h socket3102 150 44964-44979
START_RANGE client23i socket3103 140 44979-44993
START_RANGE client23j socket3104 150 44993-45008
START_RANGE client23k socket3105 140 45008-45022
START_RANGE client23l socket3106 150 45022-45037
START_RANGE client23m socket3107 140 45037-45051
START_RANGE client23n socket3108 150 45051-45066
START_RANGE client23a socket3109 140 45066-45080
START_RANGE client23b socket3110 150 45080-45095
START_RANGE client23c socket3111 140 45095-45109
START_RANGE client23d socket3112 150 45109-45124
START_RANGE client23e socket3113 140 45124-45138
START_RANGE client23f socket3114 150 45138-45153
START_RANGE client23g socket3115 140 45153-45167
START_RANGE client23h socket3116 150 45167-45182
START_RANGE client23i socket3117 140 45182-45196
START_RANGE client23j socket3118 150 45196-45211
START_RANGE client23k socket3119 140 45211-45225
START_RANGE client23l socket3120 150 45225-45240
START_RANGE client23m socket3121 140 45240-45254
START_RANGE client23n socket3122 150 45254-45269
START_RANGE client23a socket3123 140 45269-45283
START_RANGE client23b socket3124 150 45283-45298
START_RANGE client23c socket3125 140 45298-45312
START_RANGE client23d socket3126 150 45312-45327
START_RANGE client23e socket3127 140 45327-45341
START_RANGE client23f socket3128 150 45341-45356
START_RANGE client23g socket3129 140 45356-45370
START_RANGE client23h socket3130 150 45370-45385
START_RANGE client23i socket3131 140 45385-45399
START_RANGE client23j socket3132 150 45399-45414
START_RANGE client23k socket3133 140 45414-45428
START_RANGE client23l socket3134 150 45428-45443
START_RANGE client23m socket3135 140 45443-45457
START_RANGE client23n socket3136 150 45457-45472
START_RANGE client23a socket3137 140 45472-45486
START_RANGE client23b socket3138 150 45486-45501
START_RANGE client23c socket3139 140 45501-45515
START_RANGE client23d socket3140 150 45515-45530
START_RANGE client23e socket3141 140 45530-45544
START_RANGE client23f socket3142 150 45544-45559
START_RANGE client23g socket3143 140 45559-45573
START_RANGE client23h socket3144 150 45573-45588
START_RANGE client23i socket3145 140 45588-45602
START_RANGE client23j socket3146 150 45602-45617
START_RANGE client23k socket3147 140 45617-45631
START_RANGE client23l socket3148 150 45631-45646
START_RANGE client23m socket3149 140 45646-45660
START_RANGE client23n socket3150 150 45660-45675

START_RANGE client23o socket3151 140 45675-45689
START_RANGE client23p socket3152 150 45689-45704
START_RANGE client23q socket3153 140 45704-45718
START_RANGE client23r socket3154 150 45718-45733
START_RANGE client23s socket3155 140 45733-45747
START_RANGE client23t socket3156 150 45747-45762
START_RANGE client23u socket3157 140 45762-45776
START_RANGE client23v socket3158 150 45776-45791
START_RANGE client23w socket3159 140 45791-45805
START_RANGE client23x socket3160 150 45805-45820
START_RANGE client23y socket3161 140 45820-45834
START_RANGE client23z socket3162 150 45834-45849
START_RANGE client23aa socket3163 140 45849-45863
START_RANGE client23ab socket3164 150 45863-45878
START_RANGE client23o socket3165 140 45878-45892
START_RANGE client23p socket3166 150 45892-45907
START_RANGE client23q socket3167 140 45907-45921
START_RANGE client23r socket3168 150 45921-45936
START_RANGE client23s socket3169 140 45936-45950
START_RANGE client23t socket3170 150 45950-45965
START_RANGE client23u socket3171 140 45965-45979
START_RANGE client23v socket3172 150 45979-45994
START_RANGE client23w socket3173 140 45994-46008
START_RANGE client23x socket3174 150 46008-46023
START_RANGE client23y socket3175 140 46023-46037
START_RANGE client23z socket3176 150 46037-46052
START_RANGE client23aa socket3177 140 46052-46066
START_RANGE client23ab socket3178 150 46066-46081
START_RANGE client23o socket3179 140 46081-46095
START_RANGE client23p socket3180 150 46095-46110
START_RANGE client23q socket3181 140 46110-46124
START_RANGE client23r socket3182 150 46124-46139
START_RANGE client23s socket3183 140 46139-46153
```

```
START_RANGE client23t socket3184 150 46153-46168
START_RANGE client23u socket3185 140 46168-46182
START_RANGE client23v socket3186 150 46182-46197
START_RANGE client23w socket3187 140 46197-46211
START_RANGE client23x socket3188 150 46211-46226
START_RANGE client23y socket3189 140 46226-46240
START_RANGE client23z socket3190 150 46240-46255
START_RANGE client23aa socket3191 140 46255-46269
START_RANGE client23ab socket3192 150 46269-46284
START_RANGE client23o socket3193 140 46284-46298
START_RANGE client23p socket3194 150 46298-46313
START_RANGE client23q socket3195 140 46313-46327
START_RANGE client23r socket3196 150 46327-46342
START_RANGE client23s socket3197 140 46342-46356
START_RANGE client23t socket3198 150 46356-46371
START_RANGE client23u socket3199 140 46371-46385
START_RANGE client23v socket3200 150 46385-46400
START_RANGE client23w socket3201 140 46400-46414
START_RANGE client23x socket3202 150 46414-46429
START_RANGE client23y socket3203 140 46429-46443
START_RANGE client23z socket3204 150 46443-46458
START_RANGE client23aa socket3205 140 46458-46472
START_RANGE client23ab socket3206 150 46472-46487
START_RANGE client23o socket3207 140 46487-46501
START_RANGE client23p socket3208 150 46501-46516
START_RANGE client23q socket3209 140 46516-46530
START_RANGE client23r socket3210 150 46530-46545
START_RANGE client23s socket3211 140 46545-46559
START_RANGE client23t socket3212 150 46559-46574
START_RANGE client23u socket3213 140 46574-46588
START_RANGE client23v socket3214 150 46588-46603
START_RANGE client23w socket3215 140 46603-46617
START_RANGE client23x socket3216 150 46617-46632
START_RANGE client23y socket3217 140 46632-46646
START_RANGE client23z socket3218 150 46646-46661
START_RANGE client23aa socket3219 140 46661-46675
START_RANGE client23ab socket3220 150 46675-46690

START_RANGE client24a socket3221 140 46690-46704
START_RANGE client24b socket3222 150 46704-46719
START_RANGE client24c socket3223 140 46719-46733
START_RANGE client24d socket3224 150 46733-46748
START_RANGE client24e socket3225 140 46748-46762
START_RANGE client24f socket3226 150 46762-46777
START_RANGE client24g socket3227 140 46777-46791
START_RANGE client24h socket3228 150 46791-46806
START_RANGE client24i socket3229 140 46806-46820
START_RANGE client24j socket3230 150 46820-46835
START_RANGE client24k socket3231 140 46835-46849
START_RANGE client24l socket3232 150 46849-46864
START_RANGE client24m socket3233 140 46864-46878
START_RANGE client24n socket3234 150 46878-46893
START_RANGE client24a socket3235 140 46893-46907
START_RANGE client24b socket3236 150 46907-46922
START_RANGE client24c socket3237 140 46922-46936
START_RANGE client24d socket3238 150 46936-46951
START_RANGE client24e socket3239 140 46951-46965
START_RANGE client24f socket3240 150 46965-46980
START_RANGE client24g socket3241 140 46980-46994
START_RANGE client24h socket3242 150 46994-47009
START_RANGE client24i socket3243 140 47009-47023
START_RANGE client24j socket3244 150 47023-47038
START_RANGE client24k socket3245 140 47038-47052
START_RANGE client24l socket3246 150 47052-47067
START_RANGE client24m socket3247 140 47067-47081
START_RANGE client24n socket3248 150 47081-47096
START_RANGE client24a socket3249 140 47096-47110
START_RANGE client24b socket3250 150 47110-47125
START_RANGE client24c socket3251 140 47125-47139
START_RANGE client24d socket3252 150 47139-47154
START_RANGE client24e socket3253 140 47154-47168
START_RANGE client24f socket3254 150 47168-47183
START_RANGE client24g socket3255 140 47183-47197
START_RANGE client24h socket3256 150 47197-47212
START_RANGE client24i socket3257 140 47212-47226
START_RANGE client24j socket3258 150 47226-47241
START_RANGE client24k socket3259 140 47241-47255
START_RANGE client24l socket3260 150 47255-47270
START_RANGE client24m socket3261 140 47270-47284
START_RANGE client24n socket3262 150 47284-47299
START_RANGE client24a socket3263 140 47299-47313
START_RANGE client24b socket3264 150 47313-47328
START_RANGE client24c socket3265 140 47328-47342
START_RANGE client24d socket3266 150 47342-47357
START_RANGE client24e socket3267 140 47357-47371
START_RANGE client24f socket3268 150 47371-47386
START_RANGE client24g socket3269 140 47386-47400
START_RANGE client24h socket3270 150 47400-47415
START_RANGE client24i socket3271 140 47415-47429
START_RANGE client24j socket3272 150 47429-47444
START_RANGE client24k socket3273 140 47444-47458
START_RANGE client24l socket3274 150 47458-47473
START_RANGE client24m socket3275 140 47473-47487
START_RANGE client24n socket3276 150 47487-47502
START_RANGE client24a socket3277 140 47502-47516
START_RANGE client24b socket3278 150 47516-47531
START_RANGE client24c socket3279 140 47531-47545
START_RANGE client24d socket3280 150 47545-47560
START_RANGE client24e socket3281 140 47560-47574
START_RANGE client24f socket3282 150 47574-47589
START_RANGE client24g socket3283 140 47589-47603
START_RANGE client24h socket3284 150 47603-47618
START_RANGE client24i socket3285 140 47618-47632
START_RANGE client24j socket3286 150 47632-47647
START_RANGE client24k socket3287 140 47647-47661
START_RANGE client24l socket3288 150 47661-47676
START_RANGE client24m socket3289 140 47676-47690
START_RANGE client24n socket3290 150 47690-47705

START_RANGE client24o socket3291 140 47705-47719
START_RANGE client24p socket3292 150 47719-47734
START_RANGE client24q socket3293 140 47734-47748
START_RANGE client24r socket3294 150 47748-47763
START_RANGE client24s socket3295 140 47763-47777
START_RANGE client24t socket3296 150 47777-47792
START_RANGE client24u socket3297 140 47792-47806
START_RANGE client24v socket3298 150 47806-47821
START_RANGE client24w socket3299 140 47821-47835
START_RANGE client24x socket3300 150 47835-47850
START_RANGE client24y socket3301 140 47850-47864
START_RANGE client24z socket3302 150 47864-47879
START_RANGE client24aa socket3303 140 47879-47893
START_RANGE client24ab socket3304 150 47893-47908
START_RANGE client24o socket3305 140 47908-47922
START_RANGE client24p socket3306 150 47922-47937
START_RANGE client24q socket3307 140 47937-47951
START_RANGE client24r socket3308 150 47951-47966
START_RANGE client24s socket3309 140 47966-47980
START_RANGE client24t socket3310 150 47980-47995
START_RANGE client24u socket3311 140 47995-48009
START_RANGE client24v socket3312 150 48009-48024
START_RANGE client24w socket3313 140 48024-48038
START_RANGE client24x socket3314 150 48038-48053
START_RANGE client24y socket3315 140 48053-48067
START_RANGE client24z socket3316 150 48067-48082
START_RANGE client24aa socket3317 140 48082-48096
START_RANGE client24ab socket3318 150 48096-48111
START_RANGE client24o socket3319 140 48111-48125
START_RANGE client24p socket3320 150 48125-48140
START_RANGE client24q socket3321 140 48140-48154
START_RANGE client24r socket3322 150 48154-48169
START_RANGE client24s socket3323 140 48169-48183
START_RANGE client24t socket3324 150 48183-48198
START_RANGE client24u socket3325 140 48198-48212
START_RANGE client24v socket3326 150 48212-48227
START_RANGE client24w socket3327 140 48227-48241
START_RANGE client24x socket3328 150 48241-48256
START_RANGE client24y socket3329 140 48256-48270
START_RANGE client24z socket3330 150 48270-48285
START_RANGE client24aa socket3331 140 48285-48299
START_RANGE client24ab socket3332 150 48299-48314
START_RANGE client24o socket3333 140 48314-48328
START_RANGE client24p socket3334 150 48328-48343
START_RANGE client24q socket3335 140 48343-48357
START_RANGE client24r socket3336 150 48357-48372
START_RANGE client24s socket3337 140 48372-48386
START_RANGE client24t socket3338 150 48386-48401
START_RANGE client24u socket3339 140 48401-48415
START_RANGE client24v socket3340 150 48415-48430
START_RANGE client24w socket3341 140 48430-48444
START_RANGE client24x socket3342 150 48444-48459
START_RANGE client24y socket3343 140 48459-48473
START_RANGE client24z socket3344 150 48473-48488
START_RANGE client24aa socket3345 140 48488-48502
START_RANGE client24ab socket3346 150 48502-48517
START_RANGE client24o socket3347 140 48517-48531
START_RANGE client24p socket3348 150 48531-48546
START_RANGE client24q socket3349 140 48546-48560
START_RANGE client24r socket3350 150 48560-48575
START_RANGE client24s socket3351 140 48575-48589
START_RANGE client24t socket3352 150 48589-48604
START_RANGE client24u socket3353 140 48604-48618
START_RANGE client24v socket3354 150 48618-48633
START_RANGE client24w socket3355 140 48633-48647
START_RANGE client24x socket3356 150 48647-48662
START_RANGE client24y socket3357 140 48662-48676
START_RANGE client24z socket3358 150 48676-48691
START_RANGE client24aa socket3359 140 48691-48705
START_RANGE client24ab socket3360 150 48705-48720

#elif MASTER_NUM9
START_RANGE client25a socket3361 140 48720-48734
START_RANGE client25b socket3362 150 48734-48749
START_RANGE client25c socket3363 140 48749-48763
START_RANGE client25d socket3364 150 48763-48778
START_RANGE client25e socket3365 140 48778-48792
START_RANGE client25f socket3366 150 48792-48807
START_RANGE client25g socket3367 140 48807-48821
START_RANGE client25h socket3368 150 48821-48836
START_RANGE client25i socket3369 140 48836-48850
START_RANGE client25j socket3370 150 48850-48865
START_RANGE client25k socket3371 140 48865-48879
START_RANGE client25l socket3372 150 48879-48894
START_RANGE client25m socket3373 140 48894-48908
START_RANGE client25n socket3374 150 48908-48923
START_RANGE client25a socket3375 140 48923-48937
START_RANGE client25b socket3376 150 48937-48952
START_RANGE client25c socket3377 140 48952-48966
```

```
START_RANGE client25d socket3378 150 48966-48981
START_RANGE client25e socket3379 140 48981-48995
START_RANGE client25f socket3380 150 48995-49010
START_RANGE client25g socket3381 140 49010-49024
START_RANGE client25h socket3382 150 49024-49039
START_RANGE client25i socket3383 140 49039-49053
START_RANGE client25j socket3384 150 49053-49068
START_RANGE client25k socket3385 140 49068-49082
START_RANGE client25l socket3386 150 49082-49097
START_RANGE client25m socket3387 140 49097-49111
START_RANGE client25n socket3388 150 49111-49126
START_RANGE client25a socket3389 140 49126-49140
START_RANGE client25b socket3390 150 49140-49155
START_RANGE client25c socket3391 140 49155-49169
START_RANGE client25d socket3392 150 49169-49184
START_RANGE client25e socket3393 140 49184-49198
START_RANGE client25f socket3394 150 49198-49213
START_RANGE client25g socket3395 140 49213-49227
START_RANGE client25h socket3396 150 49227-49242
START_RANGE client25i socket3397 140 49242-49256
START_RANGE client25j socket3398 150 49256-49271
START_RANGE client25k socket3399 140 49271-49285
START_RANGE client25l socket3400 150 49285-49300
START_RANGE client25m socket3401 140 49300-49314
START_RANGE client25n socket3402 150 49314-49329
START_RANGE client25a socket3403 140 49329-49343
START_RANGE client25b socket3404 150 49343-49358
START_RANGE client25c socket3405 140 49358-49372
START_RANGE client25d socket3406 150 49372-49387
START_RANGE client25e socket3407 140 49387-49401
START_RANGE client25f socket3408 150 49401-49416
START_RANGE client25g socket3409 140 49416-49430
START_RANGE client25h socket3410 150 49430-49445
START_RANGE client25i socket3411 140 49445-49459
START_RANGE client25j socket3412 150 49459-49474
START_RANGE client25k socket3413 140 49474-49488
START_RANGE client25l socket3414 150 49488-49503
START_RANGE client25m socket3415 140 49503-49517
START_RANGE client25n socket3416 150 49517-49532
START_RANGE client25a socket3417 140 49532-49546
START_RANGE client25b socket3418 150 49546-49561
START_RANGE client25c socket3419 140 49561-49575
START_RANGE client25d socket3420 150 49575-49590
START_RANGE client25e socket3421 140 49590-49604
START_RANGE client25f socket3422 150 49604-49619
START_RANGE client25g socket3423 140 49619-49633
START_RANGE client25h socket3424 150 49633-49648
START_RANGE client25i socket3425 140 49648-49662
START_RANGE client25j socket3426 150 49662-49677
START_RANGE client25k socket3427 140 49677-49691
START_RANGE client25l socket3428 150 49691-49706
START_RANGE client25m socket3429 140 49706-49720
START_RANGE client25n socket3430 150 49720-49735

START_RANGE client25o socket3431 140 49735-49749
START_RANGE client25p socket3432 150 49749-49764
START_RANGE client25q socket3433 140 49764-49778
START_RANGE client25r socket3434 150 49778-49793
START_RANGE client25s socket3435 140 49793-49807
START_RANGE client25t socket3436 150 49807-49822
START_RANGE client25u socket3437 140 49822-49836
START_RANGE client25v socket3438 150 49836-49851
START_RANGE client25w socket3439 140 49851-49865
START_RANGE client25x socket3440 150 49865-49880
START_RANGE client25y socket3441 140 49880-49894
START_RANGE client25z socket3442 150 49894-49909
START_RANGE client25aa socket3443 140 49909-49923
START_RANGE client25ab socket3444 150 49923-49938
START_RANGE client25o socket3445 140 49938-49952
START_RANGE client25p socket3446 150 49952-49967
START_RANGE client25q socket3447 140 49967-49981
START_RANGE client25r socket3448 150 49981-49996
START_RANGE client25s socket3449 140 49996-50010
START_RANGE client25t socket3450 150 50010-50025
START_RANGE client25u socket3451 140 50025-50039
START_RANGE client25v socket3452 150 50039-50054
START_RANGE client25w socket3453 140 50054-50068
START_RANGE client25x socket3454 150 50068-50083
START_RANGE client25y socket3455 140 50083-50097
START_RANGE client25z socket3456 150 50097-50112
START_RANGE client25aa socket3457 140 50112-50126
START_RANGE client25ab socket3458 150 50126-50141
START_RANGE client25o socket3459 140 50141-50155
START_RANGE client25p socket3460 150 50155-50170
START_RANGE client25q socket3461 140 50170-50184
START_RANGE client25r socket3462 150 50184-50199
START_RANGE client25s socket3463 140 50199-50213
START_RANGE client25t socket3464 150 50213-50228
START_RANGE client25u socket3465 140 50228-50242
START_RANGE client25v socket3466 150 50242-50257
START_RANGE client25w socket3467 140 50257-50271
START_RANGE client25x socket3468 150 50271-50286
START_RANGE client25y socket3469 140 50286-50300
START_RANGE client25z socket3470 150 50300-50315
START_RANGE client25aa socket3471 140 50315-50329
START_RANGE client25ab socket3472 150 50329-50344
START_RANGE client25o socket3473 140 50344-50358
START_RANGE client25p socket3474 150 50358-50373
START_RANGE client25q socket3475 140 50373-50387
START_RANGE client25r socket3476 150 50387-50402
START_RANGE client25s socket3477 140 50402-50416
START_RANGE client25t socket3478 150 50416-50431
START_RANGE client25u socket3479 140 50431-50445
START_RANGE client25v socket3480 150 50445-50460
START_RANGE client25w socket3481 140 50460-50474
START_RANGE client25x socket3482 150 50474-50489
START_RANGE client25y socket3483 140 50489-50503
START_RANGE client25z socket3484 150 50503-50518
START_RANGE client25aa socket3485 140 50518-50532
START_RANGE client25ab socket3486 150 50532-50547
START_RANGE client25o socket3487 140 50547-50561
START_RANGE client25p socket3488 150 50561-50576
START_RANGE client25q socket3489 140 50576-50590
START_RANGE client25r socket3490 150 50590-50605
START_RANGE client25s socket3491 140 50605-50619
START_RANGE client25t socket3492 150 50619-50634
START_RANGE client25u socket3493 140 50634-50648
START_RANGE client25v socket3494 150 50648-50663
START_RANGE client25w socket3495 140 50663-50677
START_RANGE client25x socket3496 150 50677-50692
START_RANGE client25y socket3497 140 50692-50706
START_RANGE client25z socket3498 150 50706-50721
START_RANGE client25aa socket3499 140 50721-50735
START_RANGE client25ab socket3500 150 50735-50750

START_RANGE client26a socket3501 140 50750-50764
START_RANGE client26b socket3502 150 50764-50779
START_RANGE client26c socket3503 140 50779-50793
START_RANGE client26d socket3504 150 50793-50808
START_RANGE client26e socket3505 140 50808-50822
START_RANGE client26f socket3506 150 50822-50837
START_RANGE client26g socket3507 140 50837-50851
START_RANGE client26h socket3508 150 50851-50866
START_RANGE client26i socket3509 140 50866-50880
START_RANGE client26j socket3510 150 50880-50895
START_RANGE client26k socket3511 140 50895-50909
START_RANGE client26l socket3512 150 50909-50924
START_RANGE client26m socket3513 140 50924-50938
START_RANGE client26n socket3514 150 50938-50953
START_RANGE client26a socket3515 140 50953-50967
START_RANGE client26b socket3516 150 50967-50982
START_RANGE client26c socket3517 140 50982-50996
START_RANGE client26d socket3518 150 50996-51011
START_RANGE client26e socket3519 140 51011-51025
START_RANGE client26f socket3520 150 51025-51040
START_RANGE client26g socket3521 140 51040-51054
START_RANGE client26h socket3522 150 51054-51069
START_RANGE client26i socket3523 140 51069-51083
START_RANGE client26j socket3524 150 51083-51098
START_RANGE client26k socket3525 140 51098-51112
START_RANGE client26l socket3526 150 51112-51127
START_RANGE client26m socket3527 140 51127-51141
START_RANGE client26n socket3528 150 51141-51156
START_RANGE client26a socket3529 140 51156-51170
START_RANGE client26b socket3530 150 51170-51185
START_RANGE client26c socket3531 140 51185-51199
START_RANGE client26d socket3532 150 51199-51214
START_RANGE client26e socket3533 140 51214-51228
START_RANGE client26f socket3534 150 51228-51243
START_RANGE client26g socket3535 140 51243-51257
START_RANGE client26h socket3536 150 51257-51272
START_RANGE client26i socket3537 140 51272-51286
START_RANGE client26j socket3538 150 51286-51301
START_RANGE client26k socket3539 140 51301-51315
START_RANGE client26l socket3540 150 51315-51330
START_RANGE client26m socket3541 140 51330-51344
START_RANGE client26n socket3542 150 51344-51359
START_RANGE client26a socket3543 140 51359-51373
START_RANGE client26b socket3544 150 51373-51388
START_RANGE client26c socket3545 140 51388-51402
START_RANGE client26d socket3546 150 51402-51417
START_RANGE client26e socket3547 140 51417-51431
START_RANGE client26f socket3548 150 51431-51446
START_RANGE client26g socket3549 140 51446-51460
START_RANGE client26h socket3550 150 51460-51475
START_RANGE client26i socket3551 140 51475-51489
START_RANGE client26j socket3552 150 51489-51504
START_RANGE client26k socket3553 140 51504-51518
START_RANGE client26l socket3554 150 51518-51533
START_RANGE client26m socket3555 140 51533-51547
START_RANGE client26n socket3556 150 51547-51562
START_RANGE client26a socket3557 140 51562-51576
START_RANGE client26b socket3558 150 51576-51591
START_RANGE client26c socket3559 140 51591-51605
START_RANGE client26d socket3560 150 51605-51620
START_RANGE client26e socket3561 140 51620-51634
START_RANGE client26f socket3562 150 51634-51649
START_RANGE client26g socket3563 140 51649-51663
START_RANGE client26h socket3564 150 51663-51678
START_RANGE client26i socket3565 140 51678-51692
START_RANGE client26j socket3566 150 51692-51707
START_RANGE client26k socket3567 140 51707-51721
START_RANGE client26l socket3568 150 51721-51736
START_RANGE client26m socket3569 140 51736-51750
START_RANGE client26n socket3570 150 51750-51765

START_RANGE client26o socket3571 140 51765-51779
```

```
START_RANGE client26p socket3572 150 51779-51794
START_RANGE client26q socket3573 140 51794-51808
START_RANGE client26r socket3574 150 51808-51823
START_RANGE client26s socket3575 140 51823-51837
START_RANGE client26t socket3576 150 51837-51852
START_RANGE client26u socket3577 140 51852-51866
START_RANGE client26v socket3578 150 51866-51881
START_RANGE client26w socket3579 140 51881-51895
START_RANGE client26x socket3580 150 51895-51910
START_RANGE client26y socket3581 140 51910-51924
START_RANGE client26z socket3582 150 51924-51939
START_RANGE client26aa socket3583 140 51939-51953
START_RANGE client26ab socket3584 150 51953-51968
START_RANGE client26o socket3585 140 51968-51982
START_RANGE client26p socket3586 150 51982-51997
START_RANGE client26q socket3587 140 51997-52011
START_RANGE client26r socket3588 150 52011-52026
START_RANGE client26s socket3589 140 52026-52040
START_RANGE client26t socket3590 150 52040-52055
START_RANGE client26u socket3591 140 52055-52069
START_RANGE client26v socket3592 150 52069-52084
START_RANGE client26w socket3593 140 52084-52098
START_RANGE client26x socket3594 150 52098-52113
START_RANGE client26y socket3595 140 52113-52127
START_RANGE client26z socket3596 150 52127-52142
START_RANGE client26aa socket3597 140 52142-52156
START_RANGE client26ab socket3598 150 52156-52171
START_RANGE client26o socket3599 140 52171-52185
START_RANGE client26p socket3600 150 52185-52200
START_RANGE client26q socket3601 140 52200-52214
START_RANGE client26r socket3602 150 52214-52229
START_RANGE client26s socket3603 140 52229-52243
START_RANGE client26t socket3604 150 52243-52258
START_RANGE client26u socket3605 140 52258-52272
START_RANGE client26v socket3606 150 52272-52287
START_RANGE client26w socket3607 140 52287-52301
START_RANGE client26x socket3608 150 52301-52316
START_RANGE client26y socket3609 140 52316-52330
START_RANGE client26z socket3610 150 52330-52345
START_RANGE client26aa socket3611 140 52345-52359
START_RANGE client26ab socket3612 150 52359-52374
START_RANGE client26o socket3613 140 52374-52388
START_RANGE client26p socket3614 150 52388-52403
START_RANGE client26q socket3615 140 52403-52417
START_RANGE client26r socket3616 150 52417-52432
START_RANGE client26s socket3617 140 52432-52446
START_RANGE client26t socket3618 150 52446-52461
START_RANGE client26u socket3619 140 52461-52475
START_RANGE client26v socket3620 150 52475-52490
START_RANGE client26w socket3621 140 52490-52504
START_RANGE client26x socket3622 150 52504-52519
START_RANGE client26y socket3623 140 52519-52533
START_RANGE client26z socket3624 150 52533-52548
START_RANGE client26aa socket3625 140 52548-52562
START_RANGE client26ab socket3626 150 52562-52577
START_RANGE client26o socket3627 140 52577-52591
START_RANGE client26p socket3628 150 52591-52606
START_RANGE client26q socket3629 140 52606-52620
START_RANGE client26r socket3630 150 52620-52635
START_RANGE client26s socket3631 140 52635-52649
START_RANGE client26t socket3632 150 52649-52664
START_RANGE client26u socket3633 140 52664-52678
START_RANGE client26v socket3634 150 52678-52693
START_RANGE client26w socket3635 140 52693-52707
START_RANGE client26x socket3636 150 52707-52722
START_RANGE client26y socket3637 140 52722-52736
START_RANGE client26z socket3638 150 52736-52751
START_RANGE client26aa socket3639 140 52751-52765
START_RANGE client26ab socket3640 150 52765-52780

START_RANGE client27a socket3641 140 52780-52794
START_RANGE client27b socket3642 150 52794-52809
START_RANGE client27c socket3643 140 52809-52823
START_RANGE client27d socket3644 150 52823-52838
START_RANGE client27e socket3645 140 52838-52852
START_RANGE client27f socket3646 150 52852-52867
START_RANGE client27g socket3647 140 52867-52881
START_RANGE client27h socket3648 150 52881-52896
START_RANGE client27i socket3649 140 52896-52910
START_RANGE client27j socket3650 150 52910-52925
START_RANGE client27k socket3651 140 52925-52939
START_RANGE client27l socket3652 150 52939-52954
START_RANGE client27m socket3653 140 52954-52968
START_RANGE client27n socket3654 150 52968-52983
START_RANGE client27a socket3655 140 52983-52997
START_RANGE client27b socket3656 150 52997-53012
START_RANGE client27c socket3657 140 53012-53026
START_RANGE client27d socket3658 150 53026-53041
START_RANGE client27e socket3659 140 53041-53055
START_RANGE client27f socket3660 150 53055-53070
START_RANGE client27g socket3661 140 53070-53084
START_RANGE client27h socket3662 150 53084-53099
START_RANGE client27i socket3663 140 53099-53113
START_RANGE client27j socket3664 150 53113-53128
START_RANGE client27k socket3665 140 53128-53142
START_RANGE client27l socket3666 150 53142-53157
START_RANGE client27m socket3667 140 53157-53171
START_RANGE client27n socket3668 150 53171-53186
START_RANGE client27a socket3669 140 53186-53200
START_RANGE client27b socket3670 150 53200-53215
START_RANGE client27c socket3671 140 53215-53229
START_RANGE client27d socket3672 150 53229-53244
START_RANGE client27e socket3673 140 53244-53258
START_RANGE client27f socket3674 150 53258-53273
START_RANGE client27g socket3675 140 53273-53287
START_RANGE client27h socket3676 150 53287-53302
START_RANGE client27i socket3677 140 53302-53316
START_RANGE client27j socket3678 150 53316-53331
START_RANGE client27k socket3679 140 53331-53345
START_RANGE client27l socket3680 150 53345-53360
START_RANGE client27m socket3681 140 53360-53374
START_RANGE client27n socket3682 150 53374-53389
START_RANGE client27a socket3683 140 53389-53403
START_RANGE client27b socket3684 150 53403-53418
START_RANGE client27c socket3685 140 53418-53432
START_RANGE client27d socket3686 150 53432-53447
START_RANGE client27e socket3687 140 53447-53461
START_RANGE client27f socket3688 150 53461-53476
START_RANGE client27g socket3689 140 53476-53490
START_RANGE client27h socket3690 150 53490-53505
START_RANGE client27i socket3691 140 53505-53519
START_RANGE client27j socket3692 150 53519-53534
START_RANGE client27k socket3693 140 53534-53548
START_RANGE client27l socket3694 150 53548-53563
START_RANGE client27m socket3695 140 53563-53577
START_RANGE client27n socket3696 150 53577-53592
START_RANGE client27a socket3697 140 53592-53606
START_RANGE client27b socket3698 150 53606-53621
START_RANGE client27c socket3699 140 53621-53635
START_RANGE client27d socket3700 150 53635-53650
START_RANGE client27e socket3701 140 53650-53664
START_RANGE client27f socket3702 150 53664-53679
START_RANGE client27g socket3703 140 53679-53693
START_RANGE client27h socket3704 150 53693-53708
START_RANGE client27i socket3705 140 53708-53722
START_RANGE client27j socket3706 150 53722-53737
START_RANGE client27k socket3707 140 53737-53751
START_RANGE client27l socket3708 150 53751-53766
START_RANGE client27m socket3709 140 53766-53780
START_RANGE client27n socket3710 150 53780-53795

START_RANGE client27o socket3711 140 53795-53809
START_RANGE client27p socket3712 150 53809-53824
START_RANGE client27q socket3713 140 53824-53838
START_RANGE client27r socket3714 150 53838-53853
START_RANGE client27s socket3715 140 53853-53867
START_RANGE client27t socket3716 150 53867-53882
START_RANGE client27u socket3717 140 53882-53896
START_RANGE client27v socket3718 150 53896-53911
START_RANGE client27w socket3719 140 53911-53925
START_RANGE client27x socket3720 150 53925-53940
START_RANGE client27y socket3721 140 53940-53954
START_RANGE client27z socket3722 150 53954-53969
START_RANGE client27aa socket3723 140 53969-53983
START_RANGE client27ab socket3724 150 53983-53998
START_RANGE client27o socket3725 140 53998-54012
START_RANGE client27p socket3726 150 54012-54027
START_RANGE client27q socket3727 140 54027-54041
START_RANGE client27r socket3728 150 54041-54056
START_RANGE client27s socket3729 140 54056-54070
START_RANGE client27t socket3730 150 54070-54085
START_RANGE client27u socket3731 140 54085-54099
START_RANGE client27v socket3732 150 54099-54114
START_RANGE client27w socket3733 140 54114-54128
START_RANGE client27x socket3734 150 54128-54143
START_RANGE client27y socket3735 140 54143-54157
START_RANGE client27z socket3736 150 54157-54172
START_RANGE client27aa socket3737 140 54172-54186
START_RANGE client27ab socket3738 150 54186-54201
START_RANGE client27o socket3739 140 54201-54215
START_RANGE client27p socket3740 150 54215-54230
START_RANGE client27q socket3741 140 54230-54244
START_RANGE client27r socket3742 150 54244-54259
START_RANGE client27s socket3743 140 54259-54273
START_RANGE client27t socket3744 150 54273-54288
START_RANGE client27u socket3745 140 54288-54302
START_RANGE client27v socket3746 150 54302-54317
START_RANGE client27w socket3747 140 54317-54331
START_RANGE client27x socket3748 150 54331-54346
START_RANGE client27y socket3749 140 54346-54360
START_RANGE client27z socket3750 150 54360-54375
START_RANGE client27aa socket3751 140 54375-54389
START_RANGE client27ab socket3752 150 54389-54404
START_RANGE client27o socket3753 140 54404-54418
START_RANGE client27p socket3754 150 54418-54433
START_RANGE client27q socket3755 140 54433-54447
START_RANGE client27r socket3756 150 54447-54462
START_RANGE client27s socket3757 140 54462-54476
START_RANGE client27t socket3758 150 54476-54491
START_RANGE client27u socket3759 140 54491-54505
START_RANGE client27v socket3760 150 54505-54520
START_RANGE client27w socket3761 140 54520-54534
START_RANGE client27x socket3762 150 54534-54549
START_RANGE client27y socket3763 140 54549-54563
START_RANGE client27z socket3764 150 54563-54578
START_RANGE client27aa socket3765 140 54578-54592
START_RANGE client27ab socket3766 150 54592-54607
START_RANGE client27o socket3767 140 54607-54621
```

```
START_RANGE client27p socket3768 150 54621-54636
START_RANGE client27q socket3769 140 54636-54650
START_RANGE client27r socket3770 150 54650-54665
START_RANGE client27s socket3771 140 54665-54679
START_RANGE client27t socket3772 150 54679-54694
START_RANGE client27u socket3773 140 54694-54708
START_RANGE client27v socket3774 150 54708-54723
START_RANGE client27w socket3775 140 54723-54737
START_RANGE client27x socket3776 150 54737-54752
START_RANGE client27y socket3777 140 54752-54766
START_RANGE client27z socket3778 150 54766-54781
START_RANGE client27aa socket3779 140 54781-54795
START_RANGE client27ab socket3780 150 54795-54810

#elif MASTER_NUM10
START_RANGE client28a socket3781 140 54810-54824
START_RANGE client28b socket3782 150 54824-54839
START_RANGE client28c socket3783 140 54839-54853
START_RANGE client28d socket3784 150 54853-54868
START_RANGE client28e socket3785 140 54868-54882
START_RANGE client28f socket3786 150 54882-54897
START_RANGE client28g socket3787 140 54897-54911
START_RANGE client28h socket3788 150 54911-54926
START_RANGE client28i socket3789 140 54926-54940
START_RANGE client28j socket3790 150 54940-54955
START_RANGE client28k socket3791 140 54955-54969
START_RANGE client28l socket3792 150 54969-54984
START_RANGE client28m socket3793 140 54984-54998
START_RANGE client28n socket3794 150 54998-55013
START_RANGE client28a socket3795 140 55013-55027
START_RANGE client28b socket3796 150 55027-55042
START_RANGE client28c socket3797 140 55042-55056
START_RANGE client28d socket3798 150 55056-55071
START_RANGE client28e socket3799 140 55071-55085
START_RANGE client28f socket3800 150 55085-55100
START_RANGE client28g socket3801 140 55100-55114
START_RANGE client28h socket3802 150 55114-55129
START_RANGE client28i socket3803 140 55129-55143
START_RANGE client28j socket3804 150 55143-55158
START_RANGE client28k socket3805 140 55158-55172
START_RANGE client28l socket3806 150 55172-55187
START_RANGE client28m socket3807 140 55187-55201
START_RANGE client28n socket3808 150 55201-55216
START_RANGE client28a socket3809 140 55216-55230
START_RANGE client28b socket3810 150 55230-55245
START_RANGE client28c socket3811 140 55245-55259
START_RANGE client28d socket3812 150 55259-55274
START_RANGE client28e socket3813 140 55274-55288
START_RANGE client28f socket3814 150 55288-55303
START_RANGE client28g socket3815 140 55303-55317
START_RANGE client28h socket3816 150 55317-55332
START_RANGE client28i socket3817 140 55332-55346
START_RANGE client28j socket3818 150 55346-55361
START_RANGE client28k socket3819 140 55361-55375
START_RANGE client28l socket3820 150 55375-55390
START_RANGE client28m socket3821 140 55390-55404
START_RANGE client28n socket3822 150 55404-55419
START_RANGE client28a socket3823 140 55419-55433
START_RANGE client28b socket3824 150 55433-55448
START_RANGE client28c socket3825 140 55448-55462
START_RANGE client28d socket3826 150 55462-55477
START_RANGE client28e socket3827 140 55477-55491
START_RANGE client28f socket3828 150 55491-55506
START_RANGE client28g socket3829 140 55506-55520
START_RANGE client28h socket3830 150 55520-55535
START_RANGE client28i socket3831 140 55535-55549
START_RANGE client28j socket3832 150 55549-55564
START_RANGE client28k socket3833 140 55564-55578
START_RANGE client28l socket3834 150 55578-55593
START_RANGE client28m socket3835 140 55593-55607
START_RANGE client28n socket3836 150 55607-55622
START_RANGE client28a socket3837 140 55622-55636
START_RANGE client28b socket3838 150 55636-55651
START_RANGE client28c socket3839 140 55651-55665
START_RANGE client28d socket3840 150 55665-55680
START_RANGE client28e socket3841 140 55680-55694
START_RANGE client28f socket3842 150 55694-55709
START_RANGE client28g socket3843 140 55709-55723
START_RANGE client28h socket3844 150 55723-55738
START_RANGE client28i socket3845 140 55738-55752
START_RANGE client28j socket3846 150 55752-55767
START_RANGE client28k socket3847 140 55767-55781
START_RANGE client28l socket3848 150 55781-55796
START_RANGE client28m socket3849 140 55796-55810
START_RANGE client28n socket3850 150 55810-55825

START_RANGE client28o socket3851 140 55825-55839
START_RANGE client28p socket3852 150 55839-55854
START_RANGE client28q socket3853 140 55854-55868
START_RANGE client28r socket3854 150 55868-55883
START_RANGE client28s socket3855 140 55883-55897
START_RANGE client28t socket3856 150 55897-55912
START_RANGE client28u socket3857 140 55912-55926
START_RANGE client28v socket3858 150 55926-55941
START_RANGE client28w socket3859 140 55941-55955
START_RANGE client28x socket3860 150 55955-55970
START_RANGE client28y socket3861 140 55970-55984
START_RANGE client28z socket3862 150 55984-55999
START_RANGE client28aa socket3863 140 55999-56013
START_RANGE client28ab socket3864 150 56013-56028
START_RANGE client28o socket3865 140 56028-56042
START_RANGE client28p socket3866 150 56042-56057
START_RANGE client28q socket3867 140 56057-56071
START_RANGE client28r socket3868 150 56071-56086
START_RANGE client28s socket3869 140 56086-56100
START_RANGE client28t socket3870 150 56100-56115
START_RANGE client28u socket3871 140 56115-56129
START_RANGE client28v socket3872 150 56129-56144
START_RANGE client28w socket3873 140 56144-56158
START_RANGE client28x socket3874 150 56158-56173
START_RANGE client28y socket3875 140 56173-56187
START_RANGE client28z socket3876 150 56187-56202
START_RANGE client28aa socket3877 140 56202-56216
START_RANGE client28ab socket3878 150 56216-56231
START_RANGE client28o socket3879 140 56231-56245
START_RANGE client28p socket3880 150 56245-56260
START_RANGE client28q socket3881 140 56260-56274
START_RANGE client28r socket3882 150 56274-56289
START_RANGE client28s socket3883 140 56289-56303
START_RANGE client28t socket3884 150 56303-56318
START_RANGE client28u socket3885 140 56318-56332
START_RANGE client28v socket3886 150 56332-56347
START_RANGE client28w socket3887 140 56347-56361
START_RANGE client28x socket3888 150 56361-56376
START_RANGE client28y socket3889 140 56376-56390
START_RANGE client28z socket3890 150 56390-56405
START_RANGE client28aa socket3891 140 56405-56419
START_RANGE client28ab socket3892 150 56419-56434
START_RANGE client28o socket3893 140 56434-56448
START_RANGE client28p socket3894 150 56448-56463
START_RANGE client28q socket3895 140 56463-56477
START_RANGE client28r socket3896 150 56477-56492
START_RANGE client28s socket3897 140 56492-56506
START_RANGE client28t socket3898 150 56506-56521
START_RANGE client28u socket3899 140 56521-56535
START_RANGE client28v socket3900 150 56535-56550
START_RANGE client28w socket3901 140 56550-56564
START_RANGE client28x socket3902 150 56564-56579
START_RANGE client28y socket3903 140 56579-56593
START_RANGE client28z socket3904 150 56593-56608
START_RANGE client28aa socket3905 140 56608-56622
START_RANGE client28ab socket3906 150 56622-56637
START_RANGE client28o socket3907 140 56637-56651
START_RANGE client28p socket3908 150 56651-56666
START_RANGE client28q socket3909 140 56666-56680
START_RANGE client28r socket3910 150 56680-56695
START_RANGE client28s socket3911 140 56695-56709
START_RANGE client28t socket3912 150 56709-56724
START_RANGE client28u socket3913 140 56724-56738
START_RANGE client28v socket3914 150 56738-56753
START_RANGE client28w socket3915 140 56753-56767
START_RANGE client28x socket3916 150 56767-56782
START_RANGE client28y socket3917 140 56782-56796
START_RANGE client28z socket3918 150 56796-56811
START_RANGE client28aa socket3919 140 56811-56825
START_RANGE client28ab socket3920 150 56825-56840

START_RANGE client29a socket3921 140 56840-56854
START_RANGE client29b socket3922 150 56854-56869
START_RANGE client29c socket3923 140 56869-56883
START_RANGE client29d socket3924 150 56883-56898
START_RANGE client29e socket3925 140 56898-56912
START_RANGE client29f socket3926 150 56912-56927
START_RANGE client29g socket3927 140 56927-56941
START_RANGE client29h socket3928 150 56941-56956
START_RANGE client29i socket3929 140 56956-56970
START_RANGE client29j socket3930 150 56970-56985
START_RANGE client29k socket3931 140 56985-56999
START_RANGE client29l socket3932 150 56999-57014
START_RANGE client29m socket3933 140 57014-57028
START_RANGE client29n socket3934 150 57028-57043
START_RANGE client29a socket3935 140 57043-57057
START_RANGE client29b socket3936 150 57057-57072
START_RANGE client29c socket3937 140 57072-57086
START_RANGE client29d socket3938 150 57086-57101
START_RANGE client29e socket3939 140 57101-57115
START_RANGE client29f socket3940 150 57115-57130
START_RANGE client29g socket3941 140 57130-57144
START_RANGE client29h socket3942 150 57144-57159
START_RANGE client29i socket3943 140 57159-57173
START_RANGE client29j socket3944 150 57173-57188
START_RANGE client29k socket3945 140 57188-57202
START_RANGE client29l socket3946 150 57202-57217
START_RANGE client29m socket3947 140 57217-57231
START_RANGE client29n socket3948 150 57231-57246
START_RANGE client29a socket3949 140 57246-57260
START_RANGE client29b socket3950 150 57260-57275
START_RANGE client29c socket3951 140 57275-57289
START_RANGE client29d socket3952 150 57289-57304
START_RANGE client29e socket3953 140 57304-57318
START_RANGE client29f socket3954 150 57318-57333
START_RANGE client29g socket3955 140 57333-57347
START_RANGE client29h socket3956 150 57347-57362
START_RANGE client29i socket3957 140 57362-57376
START_RANGE client29j socket3958 150 57376-57391
START_RANGE client29k socket3959 140 57391-57405
START_RANGE client29l socket3960 150 57405-57420
START_RANGE client29m socket3961 140 57420-57434
```

```
START_RANGE client29n socket3962 150 57434-57449
START_RANGE client29a socket3963 140 57449-57463
START_RANGE client29b socket3964 150 57463-57478
START_RANGE client29c socket3965 140 57478-57492
START_RANGE client29d socket3966 150 57492-57507
START_RANGE client29e socket3967 140 57507-57521
START_RANGE client29f socket3968 150 57521-57536
START_RANGE client29g socket3969 140 57536-57550
START_RANGE client29h socket3970 150 57550-57565
START_RANGE client29i socket3971 140 57565-57579
START_RANGE client29j socket3972 150 57579-57594
START_RANGE client29k socket3973 140 57594-57608
START_RANGE client29l socket3974 150 57608-57623
START_RANGE client29m socket3975 140 57623-57637
START_RANGE client29n socket3976 150 57637-57652
START_RANGE client29a socket3977 140 57652-57666
START_RANGE client29b socket3978 150 57666-57681
START_RANGE client29c socket3979 140 57681-57695
START_RANGE client29d socket3980 150 57695-57710
START_RANGE client29e socket3981 140 57710-57724
START_RANGE client29f socket3982 150 57724-57739
START_RANGE client29g socket3983 140 57739-57753
START_RANGE client29h socket3984 150 57753-57768
START_RANGE client29i socket3985 140 57768-57782
START_RANGE client29j socket3986 150 57782-57797
START_RANGE client29k socket3987 140 57797-57811
START_RANGE client29l socket3988 150 57811-57826
START_RANGE client29m socket3989 140 57826-57840
START_RANGE client29n socket3990 150 57840-57855

START_RANGE client29o socket3991 140 57855-57869
START_RANGE client29p socket3992 150 57869-57884
START_RANGE client29q socket3993 140 57884-57898
START_RANGE client29r socket3994 150 57898-57913
START_RANGE client29s socket3995 140 57913-57927
START_RANGE client29t socket3996 150 57927-57942
START_RANGE client29u socket3997 140 57942-57956
START_RANGE client29v socket3998 150 57956-57971
START_RANGE client29w socket3999 140 57971-57985
START_RANGE client29x socket4000 150 57985-58000
START_RANGE client29y socket4001 140 58000-58014
START_RANGE client29z socket4002 150 58014-58029
START_RANGE client29aa socket4003 140 58029-58043
START_RANGE client29ab socket4004 150 58043-58058
START_RANGE client29o socket4005 140 58058-58072
START_RANGE client29p socket4006 150 58072-58087
START_RANGE client29q socket4007 140 58087-58101
START_RANGE client29r socket4008 150 58101-58116
START_RANGE client29s socket4009 140 58116-58130
START_RANGE client29t socket4010 150 58130-58145
START_RANGE client29u socket4011 140 58145-58159
START_RANGE client29v socket4012 150 58159-58174
START_RANGE client29w socket4013 140 58174-58188
START_RANGE client29x socket4014 150 58188-58203
START_RANGE client29y socket4015 140 58203-58217
START_RANGE client29z socket4016 150 58217-58232
START_RANGE client29aa socket4017 140 58232-58246
START_RANGE client29ab socket4018 150 58246-58261
START_RANGE client29o socket4019 140 58261-58275
START_RANGE client29p socket4020 150 58275-58290
START_RANGE client29q socket4021 140 58290-58304
START_RANGE client29r socket4022 150 58304-58319
START_RANGE client29s socket4023 140 58319-58333
START_RANGE client29t socket4024 150 58333-58348
START_RANGE client29u socket4025 140 58348-58362
START_RANGE client29v socket4026 150 58362-58377
START_RANGE client29w socket4027 140 58377-58391
START_RANGE client29x socket4028 150 58391-58406
START_RANGE client29y socket4029 140 58406-58420
START_RANGE client29z socket4030 150 58420-58435
START_RANGE client29aa socket4031 140 58435-58449
START_RANGE client29ab socket4032 150 58449-58464
START_RANGE client29o socket4033 140 58464-58478
START_RANGE client29p socket4034 150 58478-58493
START_RANGE client29q socket4035 140 58493-58507
START_RANGE client29r socket4036 150 58507-58522
START_RANGE client29s socket4037 140 58522-58536
START_RANGE client29t socket4038 150 58536-58551
START_RANGE client29u socket4039 140 58551-58565
START_RANGE client29v socket4040 150 58565-58580
START_RANGE client29w socket4041 140 58580-58594
START_RANGE client29x socket4042 150 58594-58609
START_RANGE client29y socket4043 140 58609-58623
START_RANGE client29z socket4044 150 58623-58638
START_RANGE client29aa socket4045 140 58638-58652
START_RANGE client29ab socket4046 150 58652-58667
START_RANGE client29o socket4047 140 58667-58681
START_RANGE client29p socket4048 150 58681-58696
START_RANGE client29q socket4049 140 58696-58710
START_RANGE client29r socket4050 150 58710-58725
START_RANGE client29s socket4051 140 58725-58739
START_RANGE client29t socket4052 150 58739-58754
START_RANGE client29u socket4053 140 58754-58768
START_RANGE client29v socket4054 150 58768-58783
START_RANGE client29w socket4055 140 58783-58797
START_RANGE client29x socket4056 150 58797-58812
START_RANGE client29y socket4057 140 58812-58826
START_RANGE client29z socket4058 150 58826-58841
START_RANGE client29aa socket4059 140 58841-58855
START_RANGE client29ab socket4060 150 58855-58870

START_RANGE client30a socket4061 140 58870-58884
START_RANGE client30b socket4062 150 58884-58899
START_RANGE client30c socket4063 140 58899-58913
START_RANGE client30d socket4064 150 58913-58928
START_RANGE client30e socket4065 140 58928-58942
START_RANGE client30f socket4066 150 58942-58957
START_RANGE client30g socket4067 140 58957-58971
START_RANGE client30h socket4068 150 58971-58986
START_RANGE client30i socket4069 140 58986-59000
START_RANGE client30j socket4070 150 59000-59015
START_RANGE client30k socket4071 140 59015-59029
START_RANGE client30l socket4072 150 59029-59044
START_RANGE client30m socket4073 140 59044-59058
START_RANGE client30n socket4074 150 59058-59073
START_RANGE client30a socket4075 140 59073-59087
START_RANGE client30b socket4076 150 59087-59102
START_RANGE client30c socket4077 140 59102-59116
START_RANGE client30d socket4078 150 59116-59131
START_RANGE client30e socket4079 140 59131-59145
START_RANGE client30f socket4080 150 59145-59160
START_RANGE client30g socket4081 140 59160-59174
START_RANGE client30h socket4082 150 59174-59189
START_RANGE client30i socket4083 140 59189-59203
START_RANGE client30j socket4084 150 59203-59218
START_RANGE client30k socket4085 140 59218-59232
START_RANGE client30l socket4086 150 59232-59247
START_RANGE client30m socket4087 140 59247-59261
START_RANGE client30n socket4088 150 59261-59276
START_RANGE client30a socket4089 140 59276-59290
START_RANGE client30b socket4090 150 59290-59305
START_RANGE client30c socket4091 140 59305-59319
START_RANGE client30d socket4092 150 59319-59334
START_RANGE client30e socket4093 140 59334-59348
START_RANGE client30f socket4094 150 59348-59363
START_RANGE client30g socket4095 140 59363-59377
START_RANGE client30h socket4096 150 59377-59392
START_RANGE client30i socket4097 140 59392-59406
START_RANGE client30j socket4098 150 59406-59421
START_RANGE client30k socket4099 140 59421-59435
START_RANGE client30l socket4100 150 59435-59450
START_RANGE client30m socket4101 140 59450-59464
START_RANGE client30n socket4102 150 59464-59479
START_RANGE client30a socket4103 140 59479-59493
START_RANGE client30b socket4104 150 59493-59508
START_RANGE client30c socket4105 140 59508-59522
START_RANGE client30d socket4106 150 59522-59537
START_RANGE client30e socket4107 140 59537-59551
START_RANGE client30f socket4108 150 59551-59566
START_RANGE client30g socket4109 140 59566-59580
START_RANGE client30h socket4110 150 59580-59595
START_RANGE client30i socket4111 140 59595-59609
START_RANGE client30j socket4112 150 59609-59624
START_RANGE client30k socket4113 140 59624-59638
START_RANGE client30l socket4114 150 59638-59653
START_RANGE client30m socket4115 140 59653-59667
START_RANGE client30n socket4116 150 59667-59682
START_RANGE client30a socket4117 140 59682-59696
START_RANGE client30b socket4118 150 59696-59711
START_RANGE client30c socket4119 140 59711-59725
START_RANGE client30d socket4120 150 59725-59740
START_RANGE client30e socket4121 140 59740-59754
START_RANGE client30f socket4122 150 59754-59769
START_RANGE client30g socket4123 140 59769-59783
START_RANGE client30h socket4124 150 59783-59798
START_RANGE client30i socket4125 140 59798-59812
START_RANGE client30j socket4126 150 59812-59827
START_RANGE client30k socket4127 140 59827-59841
START_RANGE client30l socket4128 150 59841-59856
START_RANGE client30m socket4129 140 59856-59870
START_RANGE client30n socket4130 150 59870-59885

START_RANGE client30o socket4131 140 59885-59899
START_RANGE client30p socket4132 150 59899-59914
START_RANGE client30q socket4133 140 59914-59928
START_RANGE client30r socket4134 150 59928-59943
START_RANGE client30s socket4135 140 59943-59957
START_RANGE client30t socket4136 150 59957-59972
START_RANGE client30u socket4137 140 59972-59986
START_RANGE client30v socket4138 150 59986-60001
START_RANGE client30w socket4139 140 60001-60015
START_RANGE client30x socket4140 150 60015-60030
START_RANGE client30y socket4141 140 60030-60044
START_RANGE client30z socket4142 150 60044-60059
START_RANGE client30aa socket4143 140 60059-60073
START_RANGE client30ab socket4144 150 60073-60088
START_RANGE client30o socket4145 140 60088-60102
START_RANGE client30p socket4146 150 60102-60117
START_RANGE client30q socket4147 140 60117-60131
START_RANGE client30r socket4148 150 60131-60146
START_RANGE client30s socket4149 140 60146-60160
START_RANGE client30t socket4150 150 60160-60175
START_RANGE client30u socket4151 140 60175-60189
START_RANGE client30v socket4152 150 60189-60204
START_RANGE client30w socket4153 140 60204-60218
START_RANGE client30x socket4154 150 60218-60233
START_RANGE client30y socket4155 140 60233-60247
START_RANGE client30z socket4156 150 60247-60262
```

```
START_RANGE client30aa socket4157 140 60262-60276
START_RANGE client30ab socket4158 150 60276-60291
START_RANGE client30o socket4159 140 60291-60305
START_RANGE client30p socket4160 150 60305-60320
START_RANGE client30q socket4161 140 60320-60334
START_RANGE client30r socket4162 150 60334-60349
START_RANGE client30s socket4163 140 60349-60363
START_RANGE client30t socket4164 150 60363-60378
START_RANGE client30u socket4165 140 60378-60392
START_RANGE client30v socket4166 150 60392-60407
START_RANGE client30w socket4167 140 60407-60421
START_RANGE client30x socket4168 150 60421-60436
START_RANGE client30y socket4169 140 60436-60450
START_RANGE client30z socket4170 150 60450-60465
START_RANGE client30aa socket4171 140 60465-60479
START_RANGE client30ab socket4172 150 60479-60494
START_RANGE client30o socket4173 140 60494-60508
START_RANGE client30p socket4174 150 60508-60523
START_RANGE client30q socket4175 140 60523-60537
START_RANGE client30r socket4176 150 60537-60552
START_RANGE client30s socket4177 140 60552-60566
START_RANGE client30t socket4178 150 60566-60581
START_RANGE client30u socket4179 140 60581-60595
START_RANGE client30v socket4180 150 60595-60610
START_RANGE client30w socket4181 140 60610-60624
START_RANGE client30x socket4182 150 60624-60639
START_RANGE client30y socket4183 140 60639-60653
START_RANGE client30z socket4184 150 60653-60668
START_RANGE client30aa socket4185 140 60668-60682
START_RANGE client30ab socket4186 150 60682-60697
START_RANGE client30o socket4187 140 60697-60711
START_RANGE client30p socket4188 150 60711-60726
START_RANGE client30q socket4189 140 60726-60740
START_RANGE client30r socket4190 150 60740-60755
START_RANGE client30s socket4191 140 60755-60769
START_RANGE client30t socket4192 150 60769-60784
START_RANGE client30u socket4193 140 60784-60798
START_RANGE client30v socket4194 150 60798-60813
START_RANGE client30w socket4195 140 60813-60827
START_RANGE client30x socket4196 150 60827-60842
START_RANGE client30y socket4197 140 60842-60856
START_RANGE client30z socket4198 150 60856-60871
START_RANGE client30aa socket4199 140 60871-60885
START_RANGE client30ab socket4200 150 60885-60900

#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.

   // 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

June 30th, 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 30 systems would be: 30 servers * 12,000 = $360,000. 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

Robert J. Gieringer
Worldwide Pricing Manager

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

NETGEAR®

Powered by ci

Netgear 10/100 16 Port Dual Speed Switch with Uplink Button

Model: FS116

Dealer	Dealer Product Description	Price	In Stock	
amazon.com	Netgear FS116 16-Port 10/100 Dual Speed Switch with Uplink Button	$80.99	Yes	Buy
buy.com	Netgear FS116 FS116NA	$80.98	Yes	Buy
COMPUTERS 4sure.com	Netgear FS116 10/100 16-Port Switch FS116NA	$79.95	Yes	Buy
PC Connection	Netgear Inc., FS116 10/100BaseTX 16-Port Fast Ethernet Switch + Uplink (FS116)	$109.95	Yes	Buy
PC Connection	Netgear Inc., FS116 Port 10/100BaseTX 16-port Ethernet Switch (FS116NA)	$84.95	Yes	Buy
Gateway	Netgear FS116 16-PORT 10/100BASE-TX SWITCH FS116NA	$81.95	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