IBM eServer p5 570
Model 9117-570
Using
Redhat Enterprise Linux AS 4 for IBM POWER
and
DB2 Universal Database 8.2

TPC Benchmark™ C
Full Disclosure Report

First Edition

August 8, 2005

Special Notices

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

IBM eServer pSeries

IBM eServer xSeries

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 are trademarks of the Transaction Processing Performance Council

Linux is a registered trademark of Linus Torvalds.

Linux Red Hat is a registered trademark of Red Hat, Inc.

Microsoft Windows 2000 server and COM+ are registered trademarks of Microsoft Corporation

First Edition: August 8, 2005

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, 2005. All rights reserved.

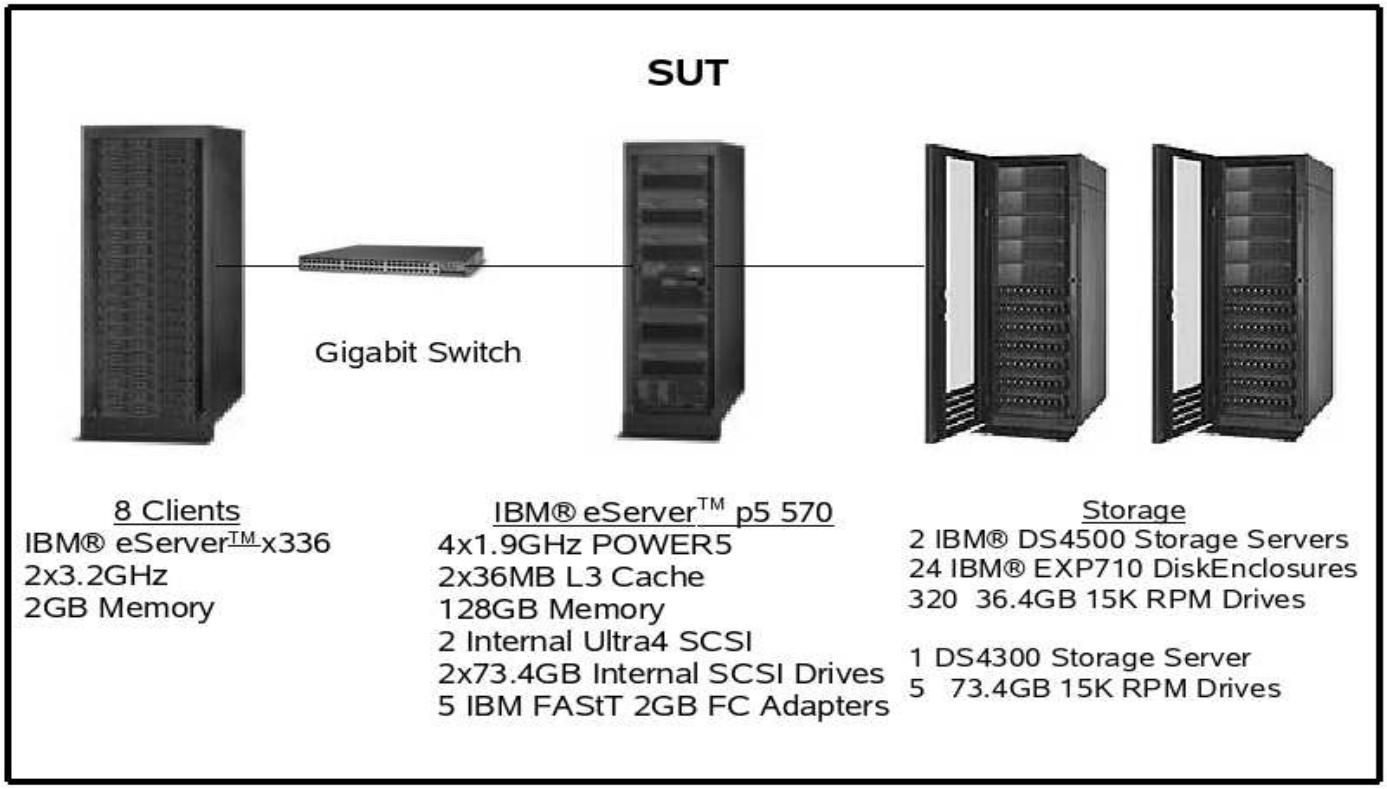
Permission is hereby granted to reproduce this document in whole or in part, provided the copyright notice printed above is set forth in full text on the title page of each item reproduced.

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

IBM	**IBM eServer p5 570 Model 9117-570**	**TPC-C Rev. 5.4** **Report Date: August 8, 2005**

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$777,707 USD	**197,669**	$3.93 USD	**Feb 7, 2006**

Processors	Database Manager	Operating System	Other Software	No. Users
4 x Server 16 x Clients	DB2 UDB 8.2	RHEL AS 4	Microsoft Visual C++ Microsoft COM+	160,000



System Components	Each of the 8 Clients		Server	
	Quantity	Description	Quantity	Description
Processors	2	3.2GHz 2MB L2 Xeon Processor	4	1.9GHz POWER5™
Memory	4	512 MB	4	32 GB
Disk Controllers	1	Ultra320 SCSI	2 5 2 1	Integrated dual Ultra3SCSI 2Gb FC Adapters IBM DS4500 Controllers for Data IBM DS4300 Controller for Log
Disk Drives	1	36 GB	320 5 2	36.4GB 15K RPM FC 73.4GB 15K RPM FC 73.4GB 10K RPM SCSI
Total Storage		288 GB		12,162 GB
Terminals	4	System Console	1	System Console

IBM eServer p5 570
Model 9117-570

TPC-C Rev. 5.4

Report Date:
August 8, 2005

Description	Part No.	Source	Unit Price	Qty	Ext Price	Maint Price
Server Hardware						
IBM eServer p5 570 24x7x4 Warr/Mnt 3 Yr	9117-570	1	2,300	1	2,300	28,362
System Drawer Enclosure	7879	1	350	1	350	0
I/O Backplane	7866	1	4,100	1	4,100	0
System Midplane	7867	1	500	1	500	0
SCSI Disk Backplane	7868	1	1,200	1	1,200	0
Power Distribution Backplane	7870	1	200	1	200	0
Serial Port Riser Card	7878	1	100	1	100	0
AC Power Supply	7888	1	800	2	1,600	0
p5 570 1.9 Processor 0/2Way	7832	1	10,900	2	21,800	0
1-way Activation for 7832	7898	1	21,800	2	43,600	0
1-way Activation for 7832 - $0 Priced Value Pack	8454	1	0	2	0	0
Media Backplane Card	7869	1	140	1	140	0
CEC Backplane	7865	1	1,200	1	1,200	0
p5 570 CPU Power Regulator	7875	1	1,100	3	3,300	0
Service Processor Card (FSP)	7881	1	650	1	650	0
IDE Slimline DVD-ROM Drive	2640	1	378	1	378	0
32GB (4x8192MB)	4492	1	103,000	4	412,000	0
DASD 73.4GB 10K rpm	3274	1	750	2	1,500	0
Line Cords	6458	1	14	2	28	0
Rack Mount Drawer Rail Kit	7164	1	168	1	168	0
Op Panel	1846	1	150	1	150	0
2 Gigabit Fibre Channel PCI-X Adapter	5716	1	2,267	5	11,335	0
Hardware Management Console	7310-C04	1	1,830	1	1,830	0
HMC Power Cord	6458	1	14	2	28	0
T541H 15" TFT Color Monitor	3637	1	625	1	625	0
Quiet Touch Keyboard	8800	1	83	1	83	0
Ethernet Cord	7801	1	12	1	12	0
HMC Warranty/Mnt. 24x7x4 3 Yr	-	1	0	1	0	1,344
Mouse with Keyboard Attachment	8841	1	62	1	62	0
Rack Model T00, 6246, 6654 24x7x4 Warr/Mnt 3 Yr	7014-T00	1	3,318	1	3,318	768
				Subtotal	**512,557**	**30,474**
Storage						
Storage Server DS4300 24x7x4 Warr/Mnt 3 Yr	1722-60U	1	14,999	1	14,999	2,499
DS4000 EXP710 Storage Expansion 24x7x4 Warr/Mnt 3 Yr	1740-1RU	1	6,000	24	144,000	18,240
SHORT WAVE SFP GBIC	2210	1	499	102	50,898	0
2GB FC, 36.4GB / 15K Drive	5212	1	1,115	320	356,800	0
2GB FC, 73.4GB / 15K Disk Module	5213	1	2,099	5	10,495	0
Storage Server DS4500 24x7x4 Warr/Mnt 3 Yr	1742-90U	1	49,900	2	99,800	2,174
Fiber Cable 1m	5601	1	79	48	3,792	0

IBM eServer p5 570
Model 9117-570

TPC-C Rev. 5.4

Report Date:
August 8, 2005

Fiber Cable 5m	5605	1	129	9	1,161	0
2GB Mini HUB	3507	1	899	4	3,596	0
Linux for POWER Host Kit	7114	1	4,000	1	4,000	0
			Subtotal		**689,541**	**22,913**
Software						
HMC Software Support 3 Yr. Support per Processor 0569	5773-RS3	1	675	1	0	675
HMC Software Support 24x7x4 3 Yr	570	1	236	1	0	236
DB2 UDB ESE 8.2 SW Lic & Mtce 12 MO		1	22,608	4	90,432	0
DB2 UDB ESE SW Mtce Renewal 2 YR		1	1,077	8	0	8,616
			Subtotal		**90,432**	**9,527**
Client Hardware and Software						
x336 3.2GHz 800MHz Xeon EM64T Express Model	8837E3U	1	4,035	8	32,280	3,600
Express IBM 36.4GB 15k U320 SCSI HS Option	30R5096	1	275	8	2,200	0
NetBay42 Standard Rack	9307-4SX	1	1,489	1	1,489	300
IBM Sleek 2-Button Mouse - (PS/2)	28L3673	1	15	1	15	0
Preferred Pro Full Size PS/2 Keyboard	31P7415	1	29	1	29	0
IBM C170 17" CRT Monitor	90P0726	1	185	1	185	90
			Subtotal		**36,198**	**3,990**
Third Party Hardware / Software						
Red Hat enterprise Linux AS 4 for POWER		3	1,499	3	4,497	0
Visual C++ Standard Edition	254-00170	2	109	1	109	0
D-Link 16 Port 10/100/1000Mbps Layer 2 Switch	674864	4	250	3	750	0
Microsoft Windows 2000 Server	C11-00821	2	738	8	5,904	0
Microsoft Problem Resolution Services		2	245	1	0	245
			Subtotal		**11,260**	**245**
				Total	**1,339,988**	**67,149**
				IBM Total System Discounts*		**(629,430)**
				Three-Year Cost of Ownership		**777,707**
					tpmC	**197,669**
					$/tpmC	**3.93**

*Discounts are based on US list prices for IBM Package of similar quantities & configurations

Pricing Sources:

1 IBM Pricing Source see FDR Appendix E

2 Microsoft Corporation 3 redhat.com 4 CDW.com

Benchmark methodologies and results independently verified by Francois Raab of InfoSizing (www.sizing.com)

The System as configured for the test will be available February 7, 2006.

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 p5 570 Model 9117-570

MQTH, computed Maximum Qualified Throughput: 197,669 tpmC

Response Times (in seconds)	90th %	Average	Maximum
New Order	0.75	0.54	1.67
Payment	0.75	0.54	3.35
Order-Status	0.76	0.55	3.05
Delivery (interactive)	0.50	0.30	1.15
Delivery (deferred)	0.36	0.22	2.95
Stock-Level	0.75	0.54	1.50
Menu	0.47	0.29	1.17

Response time delay added for emulated components was 0.1 seconds

Transaction Mix, in percent of total transactions	Percent
New Order	44.91%
Payment	43.01%
Order-Status	4.02%
Delivery	4.02%
Stock-Level	4.02%

Keying/Think Times (in seconds)	Min.	Average	Max.
New Order	18.00/0.01	18.01/12.02	18.05/120.21
Payment	3.00/0.01	3.01/12.01	3.05/120.20
Order-Status	2.00/0.01	2.01/10.01	2.04/100.10
Delivery	2.00/0.01	2.01/5.02	2.04/50.20
Stock-Level	2.00/0.01	2.01/5.02	2.04/50.20

Test Duration	
Ramp-up Time	44 minutes
Measurement interval	2 hours 30 minutes
Transactions during measurement interval (all types)	66,016,094
Ramp-down time	20 minutes

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

Table of Content

0 General Items 11
0.1. Application Code Disclosure 11
0.2. Benchmark Sponsor 11
0.3. Parameter Settings 11
0.4. Configuration Diagrams 11
1 Clause 1: Logical Data Base Design Related Items 13
1.1. Table Definitions 13
1.2. Database Organization 13
1.3. Insert and/or Delete Operations 13
1.4. Horizontal or Vertical Partitioning 13
2 Clause 2: Transaction & Terminal Profiles Related Items 14
2.1. Verification for the Random Number Generator 14
2.2. Input/Output Screens 14
2.3. Priced Terminal Features 14
2.4. Presentation Managers 14
2.5. Home and Remote Order-lines 14
2.6. New-Order Rollback Transactions 14
2.7. Number of Items per Order 15
2.8. Home and Remote Payment Transactions 15
2.9. Non-Primary Key Transactions 15
2.10. Skipped Delivery Transactions 15
2.11. Mix of Transaction Types 16
2.12. Queuing Mechanism of Delivery 16
3 Clause 3: Transaction and System Properties 17
3.1. Atomicity Requirements 17
3.2. Consistency Requirements 17
3.3. Isolation Requirements 18
3.4. Durability Requirements 18
4 Clause 4: Scaling and Data Base Population Related Items 20
4.1. Cardinality of Tables 20
4.2. Distribution of Tables and Logs 20
4.3. Data Base Model Implemented 21
4.4. Partitions/Replications Mapping 21
4.5. 60-Day Space Calculations 25
5 Clause 5: Performance Metrics and Response Time Related Items 26
5.1. Response Times 26
5.2. Keying and Think Times 26
5.3. Response Time Frequency Distribution 27
5.4. Performance Curve for Response Time versus Throughput 29
5.5. Think Time Frequency Distribution 30
5.6. Throughput versus Elapsed Time 31
5.7. Steady State Determination 31
5.8. Work Performed During Steady State 31
5.9. Measurement Interval 33
6 Clause 6: SUT, Driver, and Communication Definition Related Items 34
6.1. RTE Availability 34
6.2. Functionality and Performance of Emulated Components 34
6.3. Network Bandwidth 34
6.4. Operator Intervention 34
7 Clause 7: Pricing Related Items 35
7.1. Hardware and Programs Used 35
7.2. Three Year Cost of System Configuration 35
7.3. Availability Dates 35
7.4. Statement of tpmC and Price/Performance 35
8 Clause 9: Audit Related Items 36

Appendix - A: Client Server Code 38
Client/Terminal Handler Code 38
Transaction Code 61
Appendix - B: Tunable Parameters 106
Database Parameters 106
Transaction Monitor Parameters 108
Linux Parameters 111
Appendix - C: Database Setup Code 114
Database Creation Scripts 114
SQL Creation 138
Appendix - D: RTE Scripts 153
D.1 RTE Parameters 153
RTE Scripts 222
Appendix - E: Appendix - Pricing 231

Abstract

This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification Revision 5.4 dated April, 2005, for measurements on the IBM eServer p5 570 Model 9117-570. The software used on the IBM eServer p5 570 Model 9117-570 includes Redhat Enterprise Linux AS 4 for IBM POWER operating system, DB2 UDB 8.2 database manager. Microsoft COM+ is used as transaction manager.

IBM eServer p5 570 Model 9117-570

Company Name	**System Name**	**Data Base Software**	**Operating System Software**
IBM Corporation	IBM eServer p5 570 Model 9117-570	DB2 UDB 8.2	Redhat Enterprise Linux AS 4 for IBM POWER

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
$777,707 USD	197,669	$3.93 USD

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 5.4 in April 2005.

This is the full disclosure report for benchmark testing of the IBM eServer p5 570 Model 9117-570 and DB2 UDB 8.2 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.

0 General Items

0.1. 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.

0.2. 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.**

0.3. 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.*

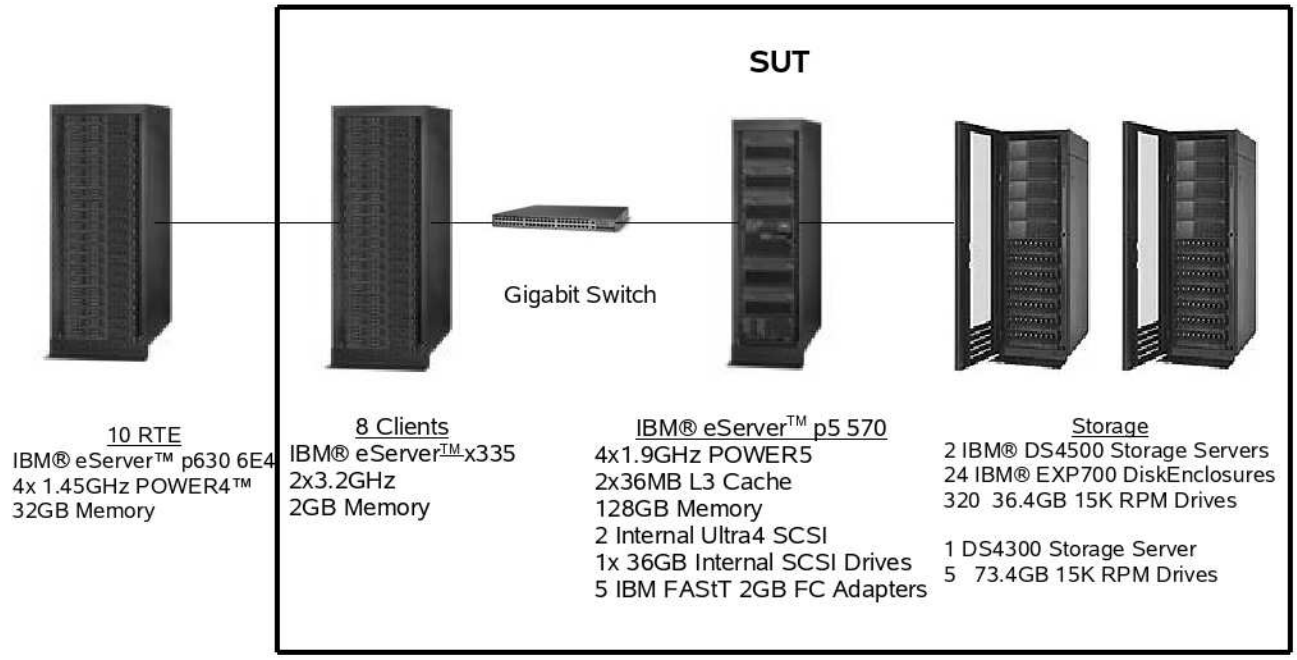
Appendix B contains the system, data base, and application parameters changed from their default values used in these TPC Benchmark™ C tests.

0.4. Configuration Diagrams

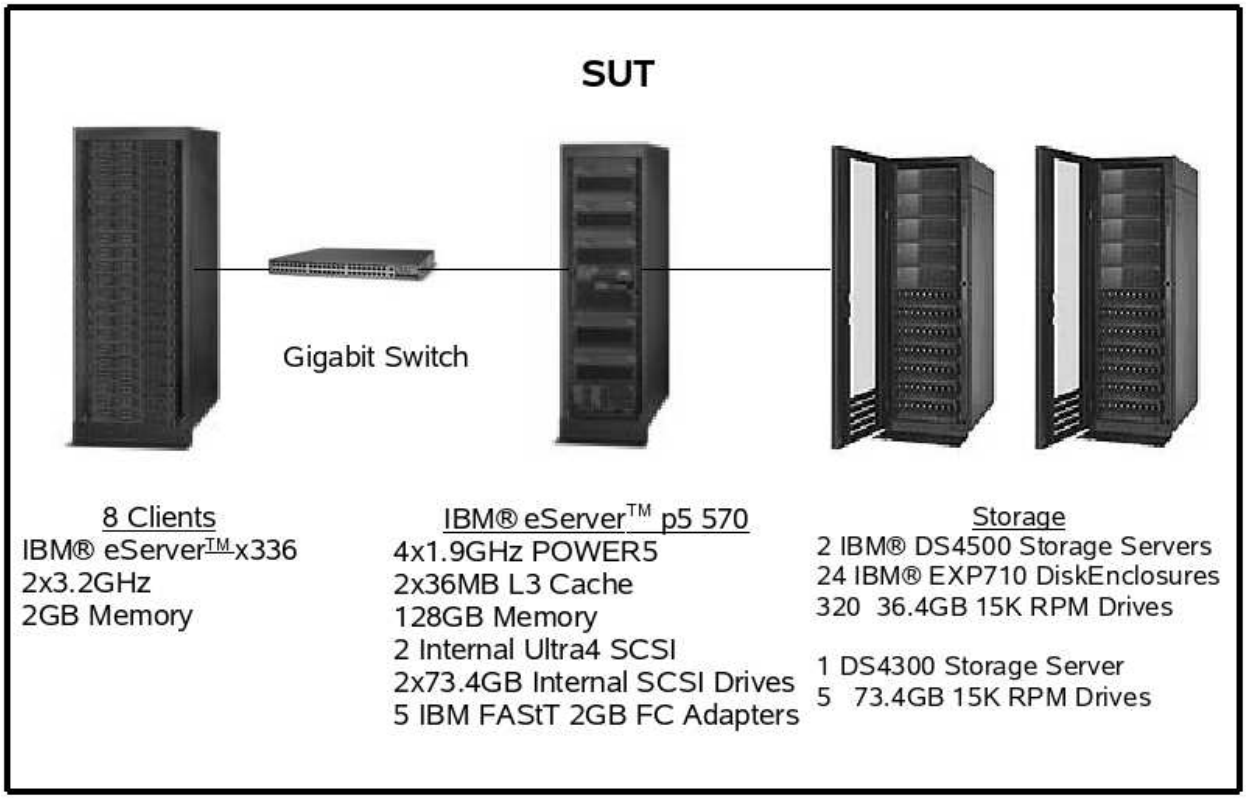
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- *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 p5 570 Model 9117-570 Benchmark Configuration



IBM eServer p5 570 Model 9117-570 Priced Configuration



The measured configuration included eight IBM eServer™ x335 client systems. They were substituted one for one in the priced configuration.

1 Clause 1: Logical Data Base Design Related Items

1.1. 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.

1.2. 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.

1.3. 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.

1.4. 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.

STOCK, CUSTOMER, ORDERS, ORDERLINE, and NEWORDER were horizontally partitioned into multiple tables.

Each (STOCK, CUSTOMER) table partition contains data associated with a range of 1680 warehouses.

Each (ORDERS, ORDERLINE, NEWORDER) table partition contains data associated with a range of 3360 warehouses.

For each partitioned table, a view was created over all table partitions to provide full transparency of data manipulation.

No tables were replicated.

2 Clause 2: Transaction & Terminal Profiles Related Items

2.1. 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.

2.2. Input/Output Screens

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

The screen layouts are now presented in HTML 1.0 web pages. Clauses 2.4.3, 2.5.3, 2.6.3, 2.7.3, and 2.8.3 of the TPC-C specifications were used as guidelines for html character placement.

2.3. 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 x335 systems, are commercially available and support all of the requirements in Clause 2.2.2.4.

2.4. 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.

2.5. Home and Remote Order-lines

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

Table 2-1 shows the percentage of home and remote transactions that occurred during the measurement period for the New-Order transactions.

2.6. 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 2-1 shows the percentage of New-Order transactions that were rolled back due to an illegal item being entered.

2.7. Number of Items per Order

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

Table 2-1 show the average number of items ordered per New-Order transaction.

2.8. Home and Remote Payment Transactions

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

Table 2-1 shows the percentage of home and remote transactions that occurred during the measurement period for the Payment transactions.

2.9. 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 2-1 shows the percentage of non-primary key accesses to the data base by the Payment and Order-Status transactions.

2.10. 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 2-1 shows the percentage of Delivery transactions missed due to a shortage of supply of rows in the NEW-ORDER table.

2.11. Mix of Transaction Types

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

Table 2-1 shows the mix percentage for each of the transaction types executed by the SUT.

New Order	**IBM eServer p5 570 Model 9117-570**
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	9.99
Payment	
Percentage of Home transactions	85.00%
Percentage of Remote transactions	15.00%
Non-Primary Key Access	
Percentage of Payment using C_LAST	60.00%
Percentage of Order-Status using C_LAST	60.02%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.91%
Payment	43.01%
Order-Status	4.02%
Delivery	4.02%
Stock-Level	4.02%

Table 2-1: Numerical Quantities for Transaction and Terminal Profiles

2.12. Queuing Mechanism of Delivery

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

The Delivery transaction was submitted to an ISAPI queue that is separate from the COM+ queue that the other transactions used. This queue is serviced by a variable amount of threads that are separate from the worker threads inside the web server. Web server threads are able to complete the on-line part of the Delivery transaction and immediately return successful queuing responses to the drivers. The threads servicing the queue are responsible for completing the deferred part of the transaction asynchronously.

3 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.

3.1. 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.

3.1.1. 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.

3.1.2. 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.

3.2. 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.

3.3. 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.

3.4. 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

3.4.1. 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 Log Disk and Log Cache
This test was conducted on a fully-scaled database. The following steps were successfully performed:

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving SUM_1.
2. A full load test was started and allowed to run for over 10 minutes.
3. One of the disks containing the transaction log was removed. Since the disk was RAID-5, DB2 continued to process the transactions successfully.
4. The test continued for at least another 5 minutes.
5. Since write-cache mirroring was enabled for the log device, the storage controller with the fiber-channel attachment to the server, which holds one copy of the mirrored cache, was removed. At this point, the server was no longer connected to the log device, and successful processing of transactions ceased.

6. The system was subsequently powered off, which removed power from all system components, including memory. The controller was reinserted into the storage subsystem holding the log device. The controller detected the cache out-of-sync condition and synchronized with the mirror cache in the other controller.

7. The disk from step 3 was replaced.

8. The system was powered back on and DB2 was allowed to recover.
9. Step 1 was performed returning the value for SUM_2. It was verified that SUM_2 was greater than SUM_1 plus the completed New_Order transactions recorded by the RTE. The additional transactions found in the database were attributed to in-flight activity at the time of the failure.
10. Consistency condition 3 was verified.

Failure of Durable Medium Containing TPC-C Database Tables:

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

1. The contents of the database were 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 12.5% of the full load.
4. A disk containing the TPC-C tables was removed, causing DB2 to report numerous errors.
5. The system was subsequently shutdown.
6. The disk was reinserted.
7. The system was powered back on.
8. The full database was restored from the backup copy in step 1.
9. DB2 was restarted and the transactions in the log were applied to the database.
10. Step 2 was performed returning SUM_2. It was verified that SUM_2 was greater than SUM_1 plus the completed New_Order transactions recorded by the RTE. The additional transactions found in the database were attributed to in-flight activity at the time of the failure.
11. Consistency condition 3 was verified.

Instantaneous Interruption and Memory Failure:

This test was conducted on a full-scaled database. The following steps were successfully performed:

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving SUM_1.
2. A full load test was started and allowed to run for over 10 minutes.
3. The system was subsequently powered off, which removed power from all system components, including memory.
4. The system was powered back on and DB2 was allowed to recover.
5. Step 1 was performed returning the value for SUM_2. It was verified that SUM_2 was greater than SUM_1 plus the completed New_Order transactions recorded by the RTE. The additional transactions found in the database were attributed to in-flight activity at the time of the failure.
6. Consistency condition 3 was verified.

4 Clause 4: Scaling and Data Base Population Related Items

4.1. 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. 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 4-1 portrays the TPC Benchmark™ C defined tables and the number of rows for each table as they were built initially.

All tables are scaled based on 16,800 warehouses, the number of active warehouses during the benchmark was 16,000

Table Name	Number of Rows
Warehouse	16,000
District	168,000
Customer	504,000,000
History	504,000,000
Order	504,000,000
New Order	151,200,000
Order Line	5,039,995,830
Stock	1,680,000,000
Item	100,000

Table 4-1: Initial Cardinality of Tables

4.2. 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 is one Logical Disk (LD) for the logs.

There is one storage adapter for the log LD.

The log LD is configured as a RAID5 disk array, with 5 physical disks. Each physical disk has a capacity of 73.4 GB. The total Raid 5 capacity available is 271.5 GB.

There are 20 Logical Disks (LD) for the tables.

There are 4 storage adapters for the tables. Each adapter is assigned 5 LDs.

Each LD is configured as a RAID0 disk array, with 16 physical disks.

There is a total of 320 data disks and each physical disk has a capacity of 36.4 GB.

Each LD is partitioned identically. Each LD contains 14 partitions, 12 of which are used for DB2 Containers. Partition are laid out on an LD as follows:

Partition#	Blocks	Container Usage
1	8001	Warehouse
2	8032	District
3	8032	Item
4		Extended partition
5	29696121	Stock

6 21687718 Customer
7 1606468 Customer Index
8 3212968 History
9 1606468 Orders
10 1606468 Order Index
11 32933218 Order Line
12 803218 New OrdersA
13 803218 New OrdersB
14 466399048 Not used for data tables

Tablespaces are laid out to use LDs as follows:

Warehouse One tablespace using 20 LDs
District One tablespace using 20 LDs
Item One tablespace using 20 LDs
Stock Ten tablespaces each using two LDs each
Customer Ten tablespaces each using two LDs each
Customer Index Ten tablespaces each using two LDs each
History One tablespace using 20 LDs
Orders Five tablespaces using 4 LDs each
Order Index Five tablespaces using 4 LDs each
Order Line Five tablespaces using 4 LDs each
New OrdersA Five tablespace using 4 LDs each
New OrdersB Five tablespace using 4 LDs each

4.3. 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 8.2. 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.

4.4. Partitions/Replications Mapping

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

The specifics of the distribution of partitioned and non-partitioned tables across the physical media can be found in Table 4-2.

Data Distribution Logical Disks (LDs) attached to FC1 storage adapter			
PARTITION	Storage Adapter	RAW Device	Assigned Tablespace
LD1 – LD5 Partition 1	FC1	raw1 –raw5	ts_wh
LD6 – LD10 Partition 1	FC2	raw6 –raw10	ts_wh
LD11 – LD15 Partition 1	FC3	raw11 –raw15	ts_wh
LD16 – LD20 Partition 1	FC4	raw16 –raw20	ts_wh
LD1 – LD5 Partition 2	FC1	raw21 –raw25	ts_dis
LD6 – LD10 Partition 2	FC2	raw26 –raw30	ts_dis
LD11 – LD15 Partition 2	FC3	raw31- raw35	ts_dis
LD16 – LD20 Partition 2	FC4	raw36 – raw40	ts_dis

LD1 – LD5 Partition 3	FC1	raw41 –raw45	ts_item
LD6 – LD10 Partition 3	FC2	raw46 –raw50	ts_item
LD11 – LD15 Partition 3	FC3	raw51- raw55	ts_item
LD16 – LD20 Partition 3	FC4	raw56 – raw60	ts_item
LD1 – LD2 Partition 5	FC1	raw61 – raw62	ts_stock_01
LD3 – LD4 Partition 5	FC1	raw63 – raw64	ts_stock_02
LD5 Partition 5	FC1	raw65	ts_stock_03
LD6 Partition 5	FC2	raw66	ts_stock_03
LD7 – LD8 Partition 5	FC2	raw67 – raw68	ts_stock_04
LD9 – LD10 Partition 5	FC2	raw69 – raw70	ts_stock_05
LD11-LD12 Partition 5	FC3	raw71 – raw 72	ts_stock_06
LD13 – LD14 Partition 5	FC3	raw73 – raw74	ts_stock_07
LD15 Partition 5	FC3	raw75	ts_stock_08
LD16 Partition 5	FC4	raw76	ts_stock_08
LD17 – LD18 Partition 5	FC4	raw77 – raw78	ts_stock_09
LD19 – LD20 Partition 5	FC4	raw79 – raw80	ts_stock_10

Data Distribution Logical Disks (LDs) attached to FC1 storage adapter			
PARTITION	Storage Adapter	RAW Device	Assigned Tablespace
LD1 – LD2 Partition 6	FC1	raw81 – raw82	ts_ customer _01
LD3 – LD4 Partition 6	FC1	raw83 – raw84	ts_ customer _02
LD5 Partition 6	FC1	raw85	ts_ customer _03
LD6 Partition 6	FC2	raw86	ts_ customer _03
LD7 – LD8 Partition 6	FC2	raw87 – raw88	ts_ customer _04
LD9 – LD10 Partition 6	FC2	raw89 – raw90	ts_ customer _05
LD11-LD12 Partition 6	FC3	raw91 – raw92	ts_ customer _06
LD13 – LD14 Partition 6	FC3	raw93 – raw94	ts_ customer _07
LD15 Partition 6	FC3	raw95	ts_ customer _08
LD16 Partition 6	FC4	raw96	ts_ customer _08
LD17 – LD18 Partition 6	FC4	raw97 – raw98	ts_ customer _09
LD19 – LD20 Partition 6	FC4	raw99 – raw100	ts_customer_10
LD1 – LD2 Partition 7	FC1	raw101 – raw102	is_ customer _01

LD3 – LD4 Partition 7	FC1	raw103 – raw104	is_ customer _02
LD5 Partition 7	FC1	raw105	is_ customer _03
LD6 Partition 7	FC2	raw106	is_ customer _03
LD7 – LD8 Partition 7	FC2	raw107 – raw108	is_ customer _04
LD9 – LD10 Partition 7	FC2	raw109 – raw110	is_ customer _05
LD11-LD12 Partition 7	FC3	raw111 – raw112	is_ customer _06
LD13 – LD14 Partition 7	FC3	raw113 – raw114	is_ customer _07
LD15 Partition 7	FC3	raw115	is_ customer _08
LD16 Partition 7	FC4	raw116	is_ customer _08
LD17 – LD18 Partition 7	FC4	raw117 – raw118	is_ customer _09
LD19 – LD20 Partition 7	FC4	raw119 – raw120	is_customer_10

Data Distribution Logical Disks (LDs) attached to FC1 storage adapter			
PARTITION	Storage Adapter	RAW Device	Assigned Tablespace
LD1 – LD5 Partition 8	FC1	raw121 –raw125	ts_history
LD6 – LD10 Partition 8	FC2	raw126 –raw130	ts_history
LD11 – LD15 Partition 8	FC3	raw131 –raw135	ts_history
LD16 – LD20 Partition 8	FC4	raw136 –raw140	ts_history
LD1 – LD4 Partition 9	FC1	raw141 –raw144	ts_order01
LD5 Partition 9	FC1	raw145	ts_order02
LD6-LD8 Partition9	FC2	raw146-raw148	ts_order02
LD9-LD10 Partition 9	FC2	raw149-raw150	ts_order03
LD11 – LD12 Partition 9	FC3	raw151 –raw152	ts_order03
LD13 – LD15 Partition 9	FC3	raw153- raw155	ts_order04
LD16 Partition 9	FC4	raw156	ts_order04
LD17-LD20 Partition 9	FC4	raw157-raw160	ts_order05
LD1 – LD4 Partition 10	FC1	raw161 –raw164	is_order01
LD5 Partition 10	FC1	raw165	is_order02
LD6-LD8 Partition 10	FC2	raw166-raw168	is_order02
LD9-LD10 Partition 10	FC2	raw169-raw170	is_order03
LD11 – LD12 Partition 10	FC3	raw171 –raw172	is_order03

LD13 – LD15 Partition 10	FC3	raw173- raw175	is_order04
LD16 Partition 10	FC4	raw176	is_order04
LD17-LD20 Partition 10	FC4	raw177-raw180	is_order05
LD1 – LD4 Partition 11	FC1	raw181 –raw184	ts_orderline_01
LD5 Partition 11	FC1	raw185	ts_orderline_02
LD6-LD8 Partition 11	FC2	raw186-raw188	ts_orderline_02
LD9-LD10 Partition 11	FC2	raw189-raw190	ts_orderline_03
LD11 – LD12 Partition 11	FC3	raw191 –raw192	ts_orderline_03
LD13 – LD15 Partition 11	FC3	raw193- raw195	ts_orderline_4
LD16 Partition 11	FC4	raw196	ts_orderline_04
LD17-LD20 Partition 11	FC4	raw197-raw200	ts_orderline_05

Data Distribution Logical Disks (LDs) attached to FC1 storage adapter			
PARTITION	Storage Adapter	RAW Device	Assigned Tablespace
LD1 – LD5 Partition 12	FC1	raw201 –raw205	ts_neworda
LD6 – LD10 Partition 12	FC2	raw206 –raw210	ts_neworda
LD11 – LD15 Partition 12	FC3	raw211 –raw215	ts_neworda
LD16 – LD20 Partition 12	FC4	raw216 –raw220	ts_neworda
LD1 – LD5 Partition 13	FC1	raw221 –raw225	ts_newordb
LD6 – LD10 Partition 13	FC2	raw226 –raw230	ts_newordb
LD11 – LD15 Partition 13	FC3	raw231- raw235	ts_newordb
LD16 – LD20 Partition 13	FC4	raw236 – raw240	ts_newordb
LD21	FC5	raw251	log

Table 4-2: IBM eServer p5 570 Model 9117-570 Data Distribution Benchmark Configuration

4.5. 60-Day Space Calculations

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

60-Day Space Computation

All data sizes in MB unless otherwise stated

Warehouses	16,800				
Measured TpmC	197,669				
Table	**Rows**	**Table**	**Index**	**5% Space**	**Total Space**
Warehouse	16,800	2	0	0	2
District	168,000	20	0	1	21
Item	100,000	10	0	1	11
Stock	1,680,000,000	546,920	0	27,346	574,266
Customer	504,000,000	393,800	24,320	20,906	439,026
New-Order	151,200,000	5,820	0	0	5,820
Orders	504,000,000	18,522	14,120	0	32,642
Order-Line	7,560,000,000	496,359	0	0	496,359
History	504,000,000	31,024	0	0	31,024

Free Space	114,536	30 Minute log Computations	
Dynamic Space	545,906	Log Written (KB)	14,054,266
Static Space	1,033,266	New-Order Txns	5,930,070
Daily Growth	102,770	Log Written per New-Order (KB)	2.37
Daily Spread	0		
Data Storage Requirement			
60 Days (MB)	7,199,473		
60 Days (GB)	7,031		
Log Storage Requirement			
8 Hours (GB)	214.45		

Disk Sizing	**Formatted**	**SUT**		**Priced**	
Disk Type	**Capacity (GB)**	**# of Disks**	**Capacity (GB)**	**# of Disks**	**Capacity (GB)**
DB FastT 36.4GB	33.40	320	10,688	320	10,688
LOG FastT RAID5 (5 x 73.4 GB)	73.40	5	271	5	271
OS SCSI 36GB	33.40	1	33	0	0
OS SCSI 74GB	73.40	0	0	2	147
Total Capacity			10,992		11,106

5 Clause 5: Performance Metrics and Response Time Related Items

5.1. 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 5-1 lists the response times and the ninetieth percentiles for each of the transaction types for the measured system.

5.2. Keying and Think Times

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

Table 5-1 lists the TPC-C keying and think times for the measured system.

Response Times	New Order	Payment	Order Status	Delivery (int./def.)	Stock Level	Menus
90 %	0.75	0.75	0.76	0.50/0.38	0.75	0.47
Average	0.54	0.54	0.55	0.30/0.23	0.54	0.29
Maximum	1.67	3.35	3.05	1.15/2.95	1.50	1.17
Think Times						
Minimum	0.01	0.01	0.01	0.01	0.01	N/A
Average	12.02	12.01	10.01	5.02	5.02	N/A
Maximum	120.21	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	18.05	3.05	2.04	2.04	2.04	N/A

Table 5-1: Think and Keying Times

5.3. Response Time Frequency Distribution

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

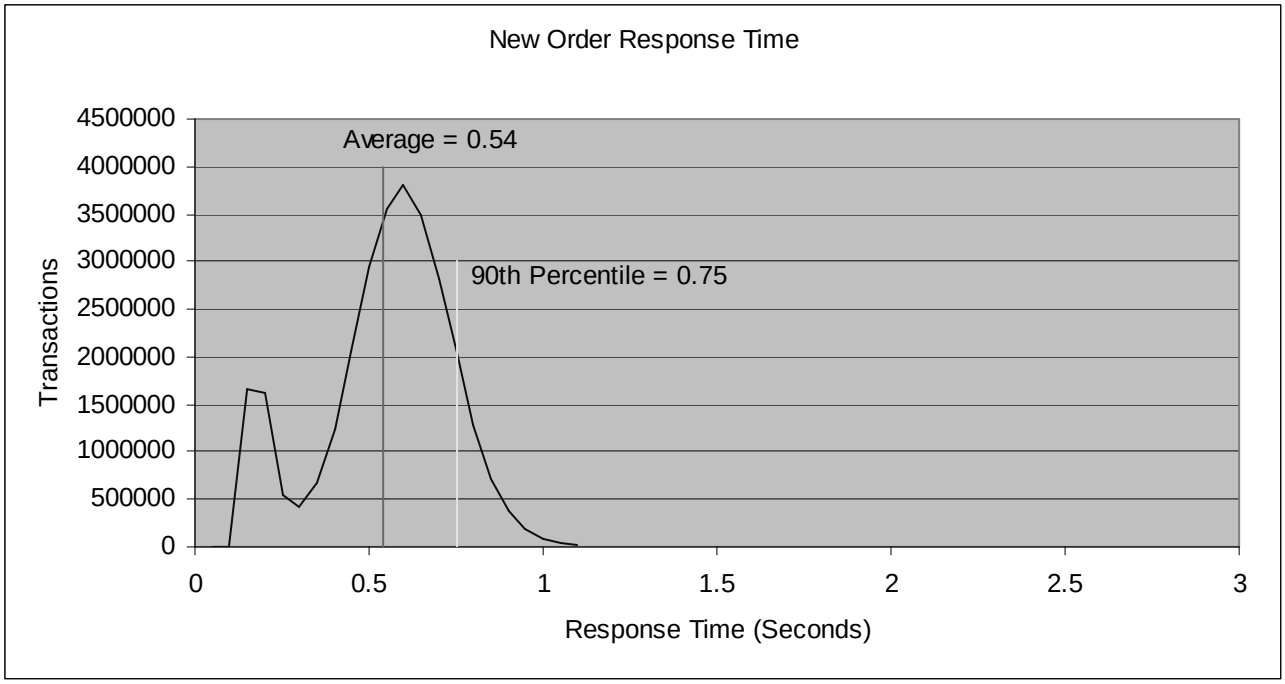


Figure 5-1: New-Order Response Time Distribution

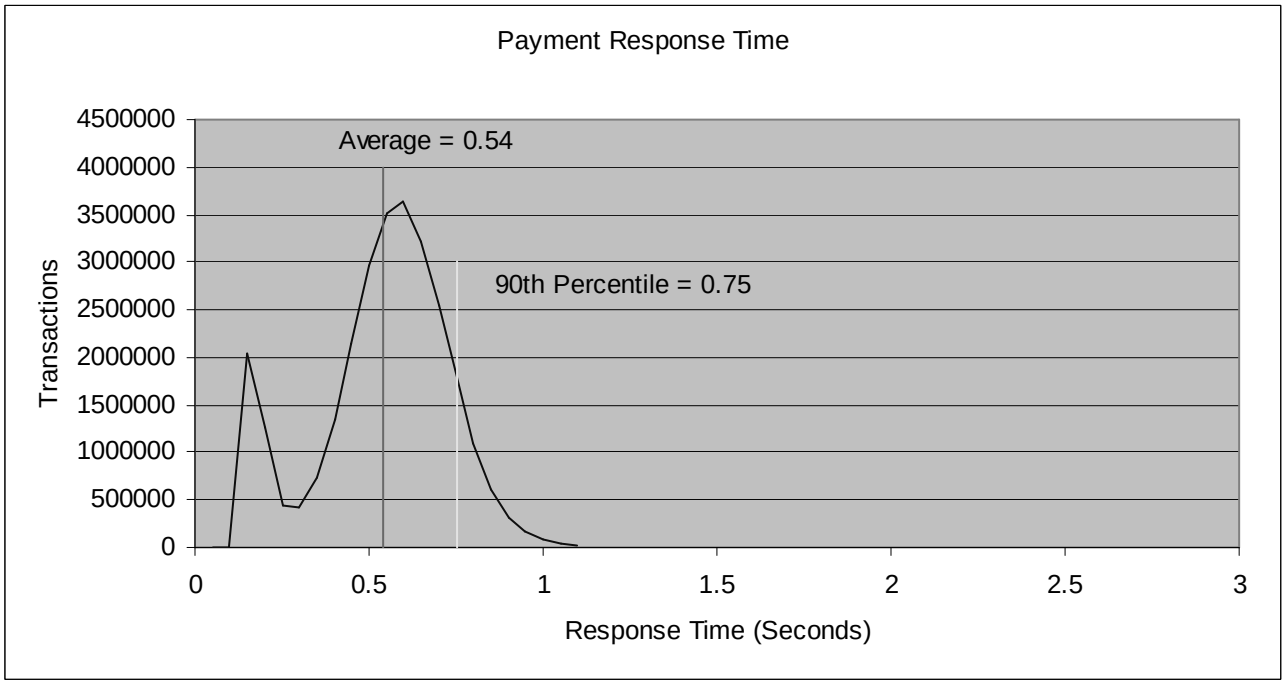


Figure 5-2: Payment Response Time Distribution

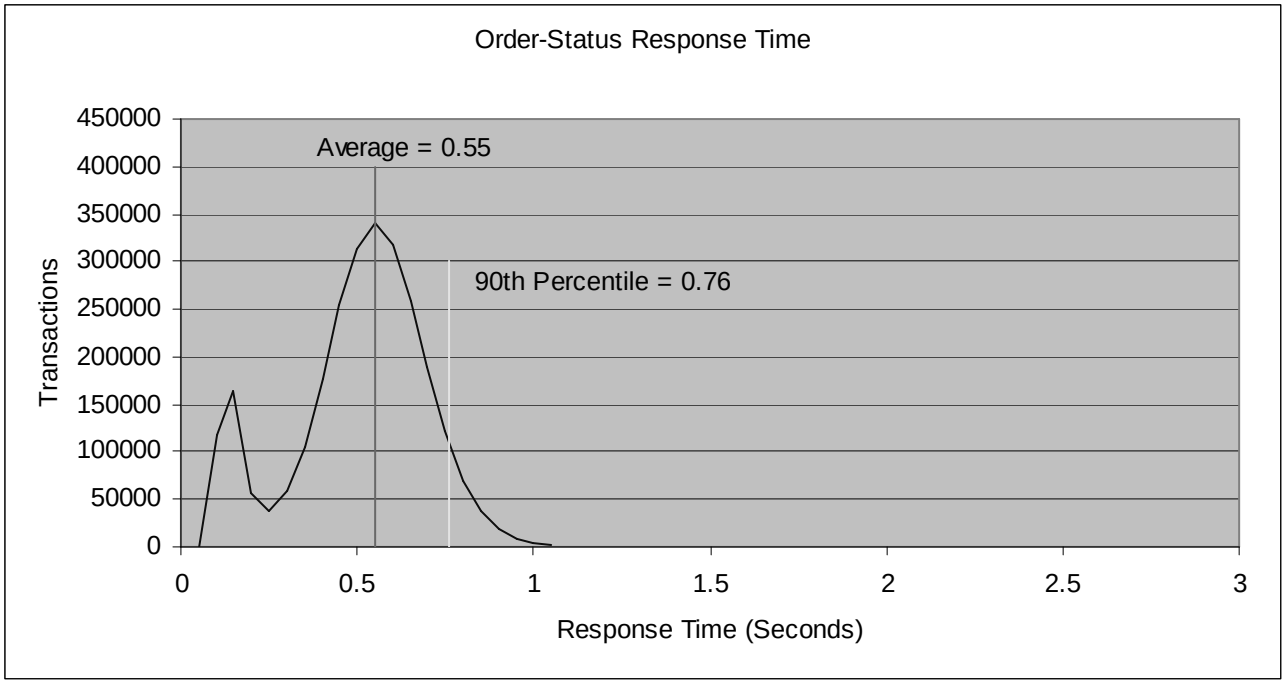


Figure 5-3: Order-Status Response Time Distribution

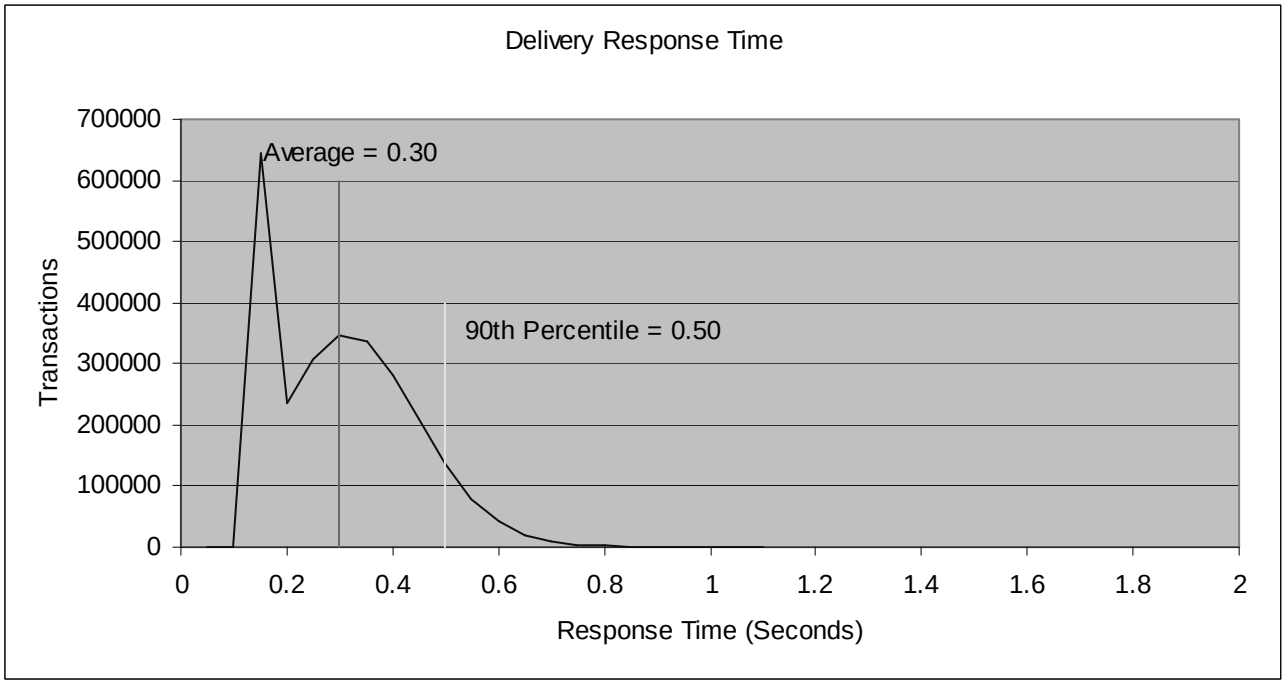


Figure 5-4: Delivery (Interactive) Response Time Distribution

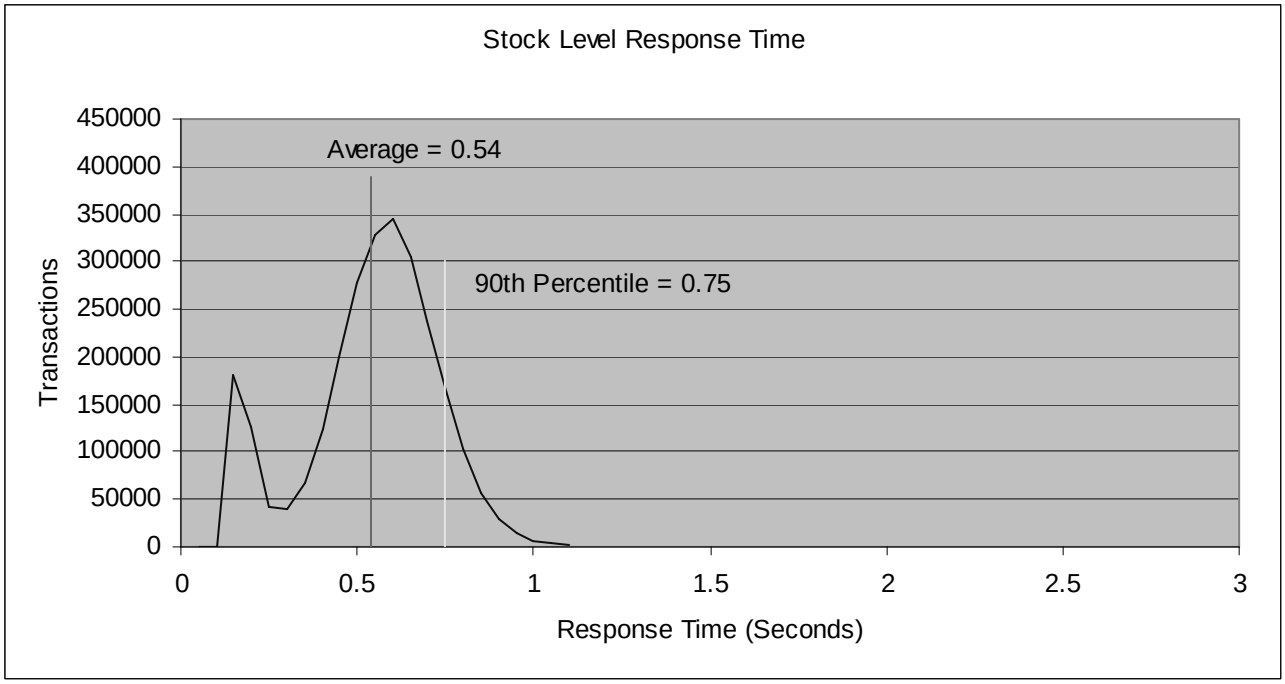


Figure 5-5: Stock Level Response Time Distribution

5.4. Performance Curve for Response Time versus Throughput

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

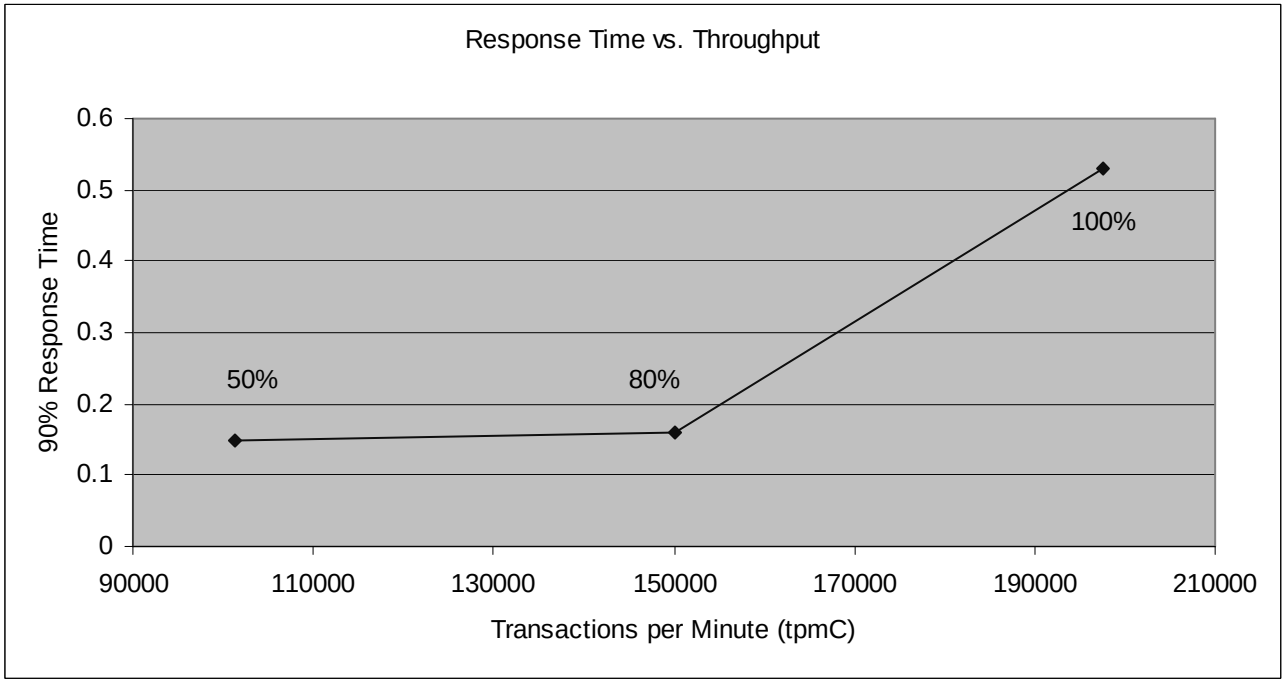


Figure 5-6: New-Order Response Time vs. Throughput

5.5. Think Time Frequency Distribution

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

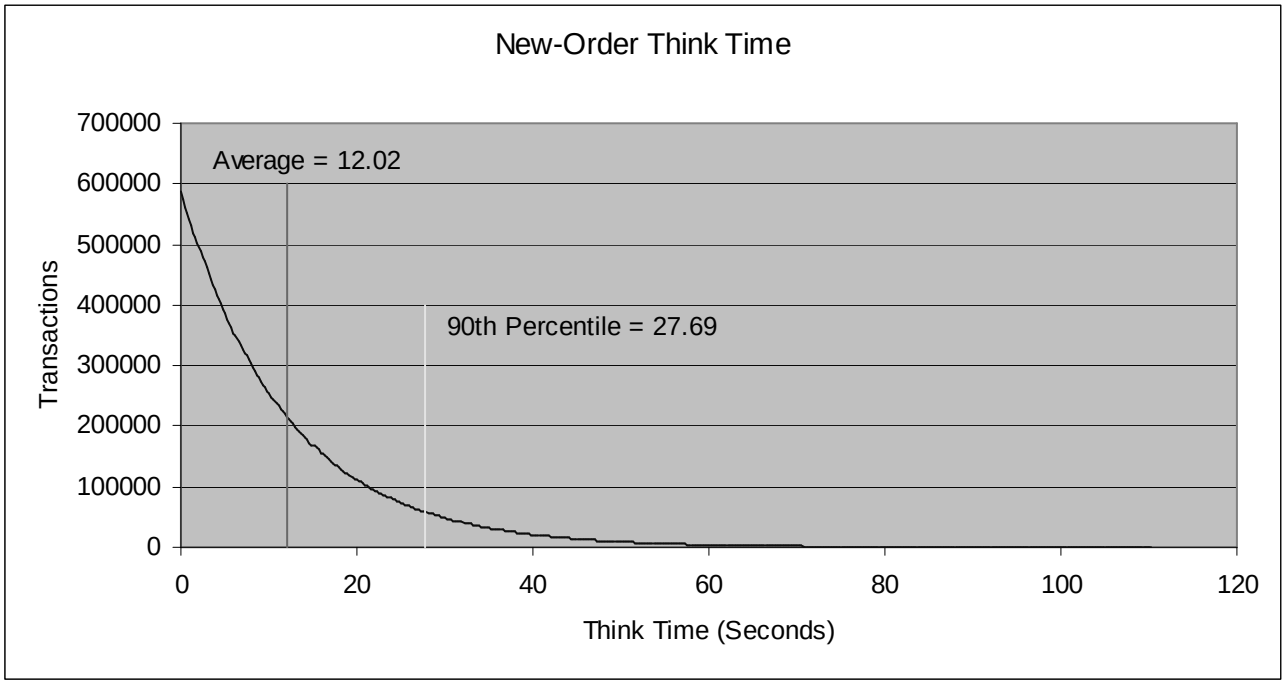


Figure 5-7: New-Order Think Time Distribution

5.6. Throughput versus Elapsed Time

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

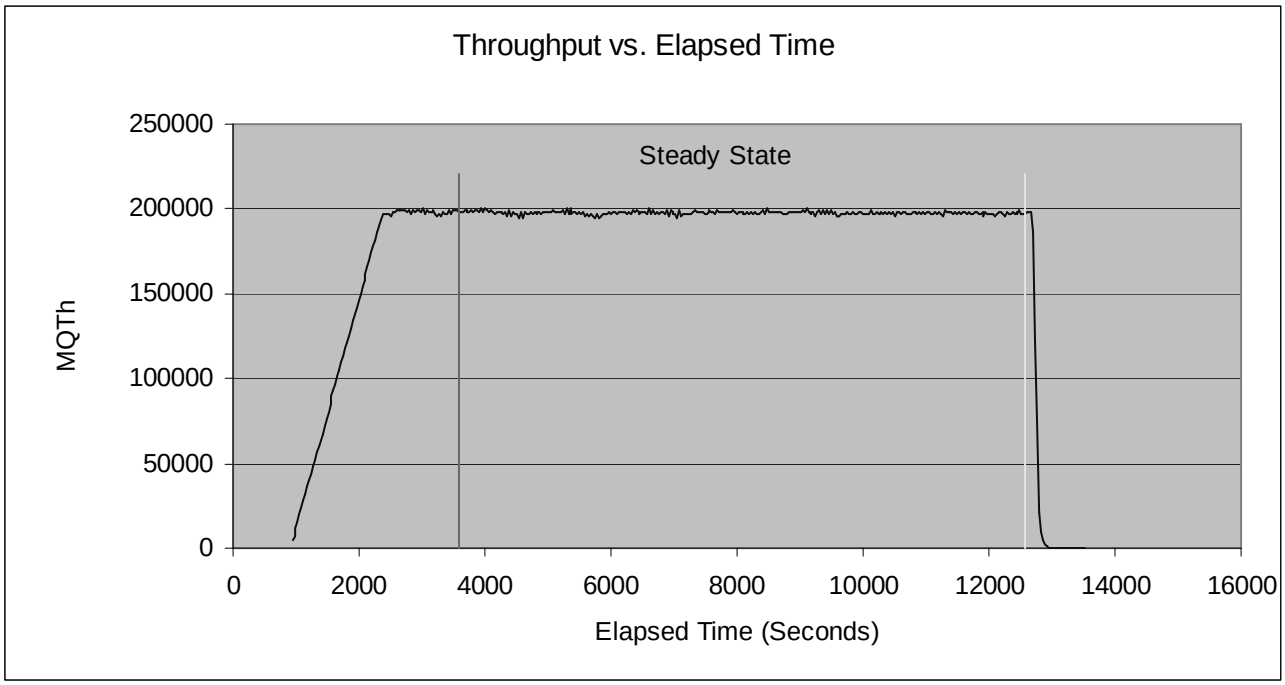


Figure 5-8: New-Order Throughput vs. Elapsed Time

5.7. 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 5-8 New-Order throughput versus Elapsed Time graph shows that the system maintained a steady state during the measurement interval

5.8. 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 2-hour 30-minute measurement interval was used to guaranty that all work normally performed during an 8-hour sustained test are included in the reported throughput.

5.8.1. Transaction Flow

Each of the 4 (non-delivery) transactions is serviced by 2 individual programs, Internet Information System 5.1 (IIS) and a Microsoft COM+ 1.0 Queued Component Server, used as the transaction manager (COM+). Both programs are running on the client system:

- The initial HTML 1.0 request is serviced by an ISAPI custom-written handler running on Internet Information System 5.1(IIS). IIS is responsible for handling all HTML requests. The web server communicates to the COM+ server through a Microsoft COM+ api interface.

- COM+ communicates with the Server system over Ethernet and handles all database operations**,** using DB2 embedded SQL calls.
  When the COM+ server boots up, it creates a configurable amount of connections to the Server (listed in application settings).
  COM+ routes the transaction and balances the load according to the options defined in the Component Services GUI for the COM+ server application and settings in the Windows 2000 Registry. The configuration file and registry variables are listed in Appendix B.2.
  At the beginning, each TPC-C user sends a pair of HTML 1.0 requests submitting the its unique warehouse and district to the IIS ISAPI handler. Upon successful validation of user's login, IIS the displays an HTML form which encapsulates the TPC-C transaction menu.
  The transaction flow is described below:
  - The TPC-C user requests the transaction type's HTML form and proceeds to generate (fill in) a GET request with the required files for the transaction.
  - IIS accepts the filled in GET request , parses, and validates all values entered by the user.
  - It then proceeds to transmit those values to the COM+ server through an transaction type specific COM+ api interface.
  - The COM+ Pool Manager receives the request and first decides if there is a connection object in the pool available to service it.
    If so, the connection is used to send the transaction request to the Server.
    If no connection is available, the request will enter a COM+ internal queue and will be serviced by the next available connection.
  - Once the connection is available to be used, a COM+ pool thread receives the transaction and calls a TPC-C back end DB2 client api to execute all database operations related to the transaction type. (All the transaction information entered on the HTML form is available in a data structure provided by the ISAPI caller).
  - The transaction is committed and the DB2 back end client returns control back to the COM pool thread.
  - COM pool thread returns control to the ISAPI caller.
    (All transaction results are inside the data structure that the ISAPI caller provided to the COM+ api in the parameter list).
  - ISAPI caller returns control to the "screen application" by doing a PUT request.

5.8.2. 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.

5.8.3. 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)

5.9. Measurement Interval

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

A 2-hour 30-minute measurement interval was used. No connections were lost during the run.

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

6.1. 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.

6.2. 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.

No components were emulated.

6.3. Network Bandwidth

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

The network between the clients and the database server was configured as 1000 MegaBits per second Full Duplex.

6.4. 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.

7 Clause 7: Pricing Related Items

7.1. 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.

7.2. 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 pricing details for this disclosure is contained in the executive summary pages. All 3rd party quotations are included at the end of this report in Appendix E. All prices are based on IBM US list prices.

Discount are based on US list prices and for similar quantities and configurations. A discount of 45.1% has been applied to all IBM hardware, software, and services based on the total value and quantities of the components of the configuration, including full payment of all components, including prepaid maintenance for three years.

7.3. 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
DB2 Universal Database 8.2	Feb 7, 2006

7.4. 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 p5 570 Model 9117-570	197,669	$777,707 USD	$3.93 USD	Feb 7, 2006

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

8 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:

Sponsors:

Raymond J. Venditti
IBM Linux Performance
11501 Burnet Road
Austin, TX 78758

Berni Schiefer
IBM DB2 Performance
8200 Warden Avenue
Markham, Ontario L6G1C7

August 8, 2005

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

Platform: IBM eServer™ p5 570 Model 9117-570 c/s
Operating system: RHEL AS 4
Database Manager: DB2 UDB 8.2
Transaction Manager: Microsoft COM+

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: IBM eServer™ p5 570				
4 x POWER5 (1.9 GHz)	128 GB main (2 x 36 MB L3 cache)	2 x 73.4 GB SCSI int. 320 x 36.4 GB FAStT 5 x 73.4 GB FAStT	0.75 Sec.	197,669.81
8 Clients: IBM eServer™ x336 (Specification for each)				
2 x Intel Xeon (3.2 GHz)	2 GB main (2 MB L2 cache/cpu)	1 x 36 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC requirements for Revision 5.4 of the benchmark.

The following verification items were given special attention:

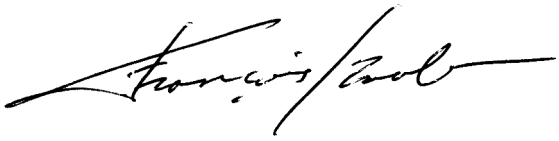
- The transactions were correctly implemented.

- 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 browser delay, 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 2.5 hours.
- Write-ahead-logging was 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 configuration included eight IBM eServer™ x335 client systems. They were substituted one for one in the priced configuration. The TPC requirements for substitution of priced clients were verified and met.

Respectfully Yours,



François Raab, President

Appendix - A: Client Server Code

Client/Terminal Handler Code

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 - 2005

## All Rights Reserved.

##

## US Government Users Restricted Rights - Use, duplication or

## disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

##############################################################################


#

# Makefile.config - NT/Win2000 Makefile Configuration

#


# Make Configuration (MSVC)

MAKE=nmake.exe


# Compiler Configuration (MSVC).

# CFLAGS_DEBUG may be set to "-Zi -Od", "-DDEBUGIT" "-Zi -Od -DDEBUGIT" or left blank

CC=cl.exe

CFLAGS_OS=-DSQLWINT -MT -DWIN32 -J -Zp8 -DREG_KIT_METHOD -DSWAP_ENDIAN

CFLAGS_OUT=/Fo

CFLAGS_DEBUG=


# Linker Configuration (MSVC)

LD_EXEC=link.exe

LD_STORP=link.exe

LDFLAGS_EXEC=

LDFLAGS_SHLIB=/DLL

LDFLAGS_STORP=$(LDFLAGS_SHLIB) /DEF:rpctpcc.def

LDFLAGS_LIB=/LIBPATH:$(TPCC_SQLLIB)\lib /LIBPATH:"C:\Program Files\Microsoft Visual Studio\VC98\Lib" db2api.lib winmm.lib

LDFLAGS_OUT=/OUT:


# Library Configuration

AR=lib.exe

ARFLAGS=

ARFLAGS_LIB=

ARFLAGS_OUT=/OUT:


# OS Commands

ERASE=del /F

ERASEDIR=rmdir /S

MOVE=MOVE

COPY=COPY


# OS File Extensions & Path Separator

OBJEXT=.obj

LIBEXT=.lib

SHLIBEXT=.dll

BINEXT=.exe

SLASH=\\

CMDSEP=&
```

Src.Cli/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 - 2005

## 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.Cli (RTE/Driver Interface)

#


!include $(TPCC_ROOT)/Makefile.config


# ##############################################################################

# Preprocessor, Compiler and Linker Flags

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

```
BND_OPTS =GRANT PUBLIC \

      MESSAGES $*.bnd.msg

PRP_OPTS =BINDFILE \

      ISOLATION RR \

      EXPLAIN ALL \

      MESSAGES $*.prep.msg \

      LEVEL $(TPCC_VERSION) \

      NOLINEMACRO



INCLUDES = -
I$(TPCC_SQLLIB)/include -
I$(TPCC_ROOT)/include



CFLAGS =  $(CFLAGS_OS)
$(INCLUDES) $(CFLAGS_DEBUG) \

      $(UOPTS) -D$(DB2EDITION) -
D$(DB2VERSION) -
D$(TPCC_SPTYPE)



OBJS =
   $(TPCC_ROOT)/Src.Common/tpcc
misc$(OBJEXT) \


   $(TPCC_ROOT)/Src.Common/tpccd
bg$(OBJEXT) \


   $(TPCC_ROOT)/Src.Common/tpccc
tx$(OBJEXT) \

      tpcccli$(OBJEXT)



LIBS =     tpcccli$(LIBEXT)



#
##############################
##############################
##########

# User Targets

#
##############################
##############################
##########



all:  connect $(OBJS) plan $(LIBS)
disconnect

   $(AR) $(ARFLAGS)
$(ARFLAGS_OUT)tpcccli$(LIBEXT)
$(OBJS) $(ARFLAGS_LIB)

   @echo "-----------------------------------
-------------------"

   @echo "Please copy lval.h,
db2tpcc.h, and tpcccli$(LIBEXT) to"

   @echo "a place where they can be
#included and linked with the"

   @echo "RTE/driver code."

   @echo "-----------------------------------
-------------------"



clean:

   - $(ERASE) *.msg *.bnd *.plan
*$(OBJEXT) *$(LIBEXT) tpcccli.c



#
##############################
##############################
##########

# Helper Targets

#
##############################
##############################
##########



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 TPCCCLI
-g -# 0 -o TPCCCLI.exfmt.plan

   - db2expln -d $(TPCC_DBNAME) -c
$(TPCC_SCHEMA) -p TPCCCLI -s 0 -
g -o TPCCCLI.expln.plan



rebind:  connect

   db2 bind tpcccli.bnd $(BND_OPTS)
QUERYOPT 7

#
##############################
##############################
##########

# Build Rules

#
##############################
##############################
##########


.SUFFIXES:

.SUFFIXES: $(OBJEXT) .c .sqc


tpcccli.c:

   @echo "Prepping $*.sqc"

   -db2 prep $*.sqc $(PRP_OPTS)
ISOLATION RR

   @echo "Binding $*.bnd"

   db2 bind $*.bnd $(BND_OPTS)
QUERYOPT 7


#
##############################
##############################
##########

# Dependencies

#
##############################
##############################
##########


# Client Library:

tpcccli$(LIBEXT): $(OBJS)


# Source

tpcc_all_sql$(OBJEXT):
   tpcc_all_sql.c


# Headers

tpcc_all_sql.c:
   $(TPCC_ROOT)/include/db2tpcc.h
$(TPCC_ROOT)/include/lval.h
```

Src.Cli/tpcccli.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 - 2005
** 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"

// ------------------------------------------------
---------------
// 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 clientRc = TRAN_OK ;

   int itemIndex = 0 ;
   int actualItemIndex = 0 ;

   /* Create Timestamp */
   in_neword->s_O_ENTRY_D_time =
time(NULL) ;

   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 ;

   // Sort the original array

   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++ )
   {
         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_SU
PPLY_W_ID = in_neword-
>in_item[ itemIndex ].s_OL_SUPPLY_
W_ID ;
         in_neword-
>in_item[ actualItemIndex ].s_OL_QU
ANTITY    = in_neword-
>in_item[ itemIndex ].s_OL_QUANTIT
Y ;
   }

   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 */

#ifdef SWAP_ENDIAN
   for (itemIndex=0;
itemIndex<in_neword->s_O_OL_CNT;
itemIndex++)
   {
      SWAP_BYTE(in_neword-
>in_item[ itemIndex ].s_OL_I_ID);
      SWAP_BYTE(in_neword-
>in_item[ itemIndex ].s_OL_SUPPLY_
W_ID);
      SWAP_BYTE(in_neword-
>in_item[ itemIndex ].s_OL_QUANTIT
Y);
   }
   SWAP_BYTE(in_neword-
>s_O_ENTRY_D_time);
   SWAP_BYTE(in_neword->s_C_ID);
   SWAP_BYTE(in_neword->s_W_ID);
   SWAP_BYTE(in_neword->s_D_ID);
   SWAP_BYTE(in_neword-
>s_O_OL_CNT);
   SWAP_BYTE(in_neword-
>s_all_local);
   SWAP_BYTE(in_neword-
>duplicate_items);
#endif  //SWAP_ENDIAN

   EXEC SQL CALL news
( :*pHostvarInput, :*pHostvarOutput );

#ifdef SWAP_ENDIAN
   SWAP_BYTE(in_neword-
>s_O_ENTRY_D_time);
   SWAP_BYTE(in_neword->s_C_ID);
   SWAP_BYTE(in_neword->s_W_ID);
   SWAP_BYTE(in_neword->s_D_ID);
   SWAP_BYTE(in_neword-
>s_O_OL_CNT);
   SWAP_BYTE(in_neword-
>s_all_local);
   SWAP_BYTE(in_neword-
>duplicate_items);
```

```
   for (itemIndex=0;
itemIndex<in_neword->s_O_OL_CNT;
itemIndex++)
   {
      SWAP_BYTE(in_neword-
>in_item[ itemIndex ].s_OL_I_ID);
      SWAP_BYTE(in_neword-
>in_item[ itemIndex ].s_OL_SUPPLY_
W_ID);
      SWAP_BYTE(in_neword-
>in_item[ itemIndex ].s_OL_QUANTIT
Y);
   }

   SWAP_BYTE(neword-
>s_O_ENTRY_D_time);
   SWAP_BYTE(neword->s_W_TAX);
   SWAP_BYTE(neword->s_D_TAX);
   SWAP_BYTE(neword-
>s_C_DISCOUNT);
   SWAP_BYTE(neword-
>s_total_amount);
   SWAP_BYTE(neword->s_O_ID);
   SWAP_BYTE(neword-
>s_O_OL_CNT);
   SWAP_BYTE(neword-
>s_transtatus);
   SWAP_BYTE(neword->deadlocks);
   for (itemIndex=0;
itemIndex<in_neword->s_O_OL_CNT;
itemIndex++)
   {
      SWAP_BYTE(neword-
>item[ itemIndex ].s_I_PRICE);
      SWAP_BYTE(neword-
>item[ itemIndex ].s_OL_AMOUNT);
      SWAP_BYTE(neword-
>item[ itemIndex ].s_S_QUANTITY);
   }
#endif  //SWAP_ENDIAN

   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 ; // from input , not
output
            itemIndex++
          )
      {
         if ( neword-
>item[ itemIndex ].s_I_PRICE > 0 )  //
A zero price signifies a bad item
         {
            neword-
>item[ itemIndex ].s_OL_AMOUNT =
neword->item[ itemIndex ].s_I_PRICE
*

in_neword-
>in_item[ itemIndex ].s_OL_QUANTIT
Y ;  // reference input value

            neword->s_total_amount +=
neword-
>item[ itemIndex ].s_OL_AMOUNT ;

         }
      }

      // s_total_amount gets cast
implicitly to a double to do the
arithmetic,
      // and then cast back to a sqlint32.
      neword->s_total_amount *= factor;
   }
   else
   {
      sqlerror( NEWORD_SQL, "NEW",
__FILE__, __LINE__, &sqlca) ;
      neword->s_transtatus =
FATAL_SQLERROR ;
      clientRc = FATAL_SQLERROR ;
   }

   /* Update Output Structure with
Timestamp */
   neword->s_O_ENTRY_D_time =
in_neword->s_O_ENTRY_D_time ;

#ifdef DEBUGIT
   new_debug(neword, in_neword,
"Client after SP call");
#endif /* DEBUGIT */

   if (neword->s_transtatus <=
FATAL_SQLERROR)
   {
      new_debug(neword, in_neword,
"NEW failed");
      clientRc = FATAL_SQLERROR ;
   }

   if (neword->s_transtatus ==
INVALID_ITEM)
   {
      clientRc = INVALID_ITEM ;
   }

   return ( clientRc ) ;
}

// ---------------------------------------------------
---------------
// Payment CLIENT
// ---------------------------------------------------
---------------

int payment_sql  (   struct
in_payment_struct * in_payment
                  , struct out_payment_struct
* payment    )
{
   struct sqlca sqlca ;

   int clientRc = TRAN_OK ;

   EXEC SQL BEGIN DECLARE
SECTION;

      // Inputs
      sqlint64 h_amount ;
      sqlint32 in_c_id ;

      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 ;
      sqlint32  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->deadlocks = -1 ;
   payment->s_transtatus = TRAN_OK ;

   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 SQL call");
#endif /* DEBUGIT */

   // Create c_data_prefix strings and
copy some elements from
   // in -> out struct outside of retry_tran
loop

   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) ) ;

      // Setup the input c_last varchar
      c_last_input.len =
strlen( in_payment->s_C_LAST ) ;
      memcpy( c_last_input.data ,
in_payment->s_C_LAST ,
c_last_input.len ) ;

      // Copy to the output structure
      memcpy( payment->s_C_LAST ,
in_payment->s_C_LAST,
sizeof( payment->s_C_LAST ) ) ;

   } else {

      // Copy c_id to the output structure
      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_sin
ce , :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_prefi
x_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_middl
e , :c_street_1 , :c_street_2 , :c_city , :
c_state
                , :c_zip , :c_phone , :c_sin
ce , :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 ;

   }

   /* Update Output Structure with
Timestamp */
   payment->s_H_DATE_time =
in_payment->s_H_DATE_time ;

#ifdef DEBUGIT
   pay_debug(payment, in_payment,
"Client after SQL call");
#endif /* DEBUGIT */

   if ( sqlca.sqlcode != 0 )
   {
      DLCHK( retry_tran ) ;

      sqlerror( PAYMENT_SQL , "PAY" ,
__FILE__, __LINE__ , &sqlca) ;
      payment->s_transtatus =
FATAL_SQLERROR ;
      clientRc = FATAL_SQLERROR ;

      pay_debug( payment, in_payment,
"PAY failed" ) ;

      EXEC SQL ROLLBACK WORK ;

      if ( sqlca.sqlcode != 0 )
      {
         sqlerror( PAYMENT_SQL,
"ROLLBACK FAILED", __FILE__,
__LINE__, &sqlca ) ;
      }
   }

   return ( clientRc ) ;
}

// ---------------------------------------------
---------------
// 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 clientRc = TRAN_OK ;
   int itemIndex = 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 */

#ifdef SWAP_ENDIAN
   SWAP_BYTE(in_ordstat->s_C_ID);
   SWAP_BYTE(in_ordstat->s_W_ID);
   SWAP_BYTE(in_ordstat->s_D_ID);
#endif  //SWAP_ENDIAN

   EXEC SQL CALL ords
( :*in_ord, :*out_ord ) ;

#ifdef SWAP_ENDIAN
   SWAP_BYTE(in_ordstat->s_C_ID);
   SWAP_BYTE(in_ordstat->s_W_ID);
   SWAP_BYTE(in_ordstat->s_D_ID);

   SWAP_BYTE(ordstat-
>s_C_BALANCE);
   SWAP_BYTE(ordstat-
>s_O_ENTRY_D_time);
   SWAP_BYTE(ordstat->s_C_ID);
   SWAP_BYTE(ordstat->s_O_ID);
   SWAP_BYTE(ordstat-
>s_O_CARRIER_ID);
   SWAP_BYTE(ordstat->s_ol_cnt);
   SWAP_BYTE(ordstat->s_transtatus);
   SWAP_BYTE(ordstat->deadlocks);
   for (itemIndex=0; itemIndex<ordstat-
>s_ol_cnt; itemIndex++)
   {
      SWAP_BYTE(ordstat-
>item[ itemIndex ].s_OL_DELIVERY_
D_time);
      SWAP_BYTE(ordstat-
>item[ itemIndex ].s_OL_AMOUNT);
      SWAP_BYTE(ordstat-
>item[ itemIndex ].s_OL_I_ID);
      SWAP_BYTE(ordstat-
>item[ itemIndex ].s_OL_SUPPLY_W_
ID);
      SWAP_BYTE(ordstat-
>item[ itemIndex ].s_OL_QUANTITY);
   }
#endif  //SWAP_ENDIAN

   if ( sqlca.sqlcode == 0 )
   {
      // Propogate the field we already
knew into the output structure
      // 60% of the time, we already new
c_last (input c_id is 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",
__FILE__, __LINE__, &sqlca) ;
      ordstat->s_transtatus =
FATAL_SQLERROR ;
      clientRc = FATAL_SQLERROR ;
   }

#ifdef DEBUGIT
   ord_debug(ordstat, in_ordstat,
"Client after SP call");
#endif /* DEBUGIT */

   if ( ordstat->s_transtatus <=
FATAL_SQLERROR )
   {
      ord_debug(ordstat, in_ordstat,
"ORD failed");
      clientRc = FATAL_SQLERROR ;
```

```
   }

   return ( clientRc ) ;
}

// ---------------------------------------------------
---------------
// 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 clientRc = TRAN_OK ;
   int orderIndex = 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 */

#ifdef SWAP_ENDIAN
   SWAP_BYTE(in_delivery-
>s_O_DELIVERY_D_time);
   SWAP_BYTE(in_delivery->s_W_ID);
   SWAP_BYTE(in_delivery-
>s_O_CARRIER_ID);
#endif  //SWAP_ENDIAN

   EXEC SQL CALL dels
( :*in_del, :*out_del ) ;

#ifdef SWAP_ENDIAN
   SWAP_BYTE(in_delivery-
>s_O_DELIVERY_D_time);
   SWAP_BYTE(in_delivery->s_W_ID);
   SWAP_BYTE(in_delivery-
>s_O_CARRIER_ID);

   for (orderIndex=0; orderIndex<10;
orderIndex++) {
      SWAP_BYTE(delivery-
>s_O_ID[ orderIndex ]);
   }
   SWAP_BYTE(delivery-
>s_transtatus);
   SWAP_BYTE(delivery->deadlocks);
#endif  //SWAP_ENDIAN

#ifdef DEBUGIT
   del_debug(delivery, in_delivery,
"Client after SP call");
#endif /* DEBUGIT */

   if ( sqlca.sqlcode != 0 )
   {
      sqlerror( DELIVERY_SQL, "DEL",
__FILE__, __LINE__, &sqlca) ;
      delivery->s_transtatus =
FATAL_SQLERROR ;
      clientRc = FATAL_SQLERROR ;
   }

   if ( delivery->s_transtatus <=
FATAL_SQLERROR )
   {
      del_debug(delivery, in_delivery,
"DEL failed");
      clientRc = FATAL_SQLERROR ;
   }

   return ( clientRc ) ;
}

// ---------------------------------------------------
---------------
// 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 clientRc = TRAN_OK ;

   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 ;
   stocklev->s_transtatus = TRAN_OK ;

#ifdef DEBUGIT
   stk_debug(stocklev, in_stocklev,
"Client before SQL call");
#endif /* DEBUGIT */

retry_tran:

   stocklev->deadlocks ++ ;

/*
   EXEC SQL

      SELECT
ITEMS_BELOW_THRESHOLD

        INTO :low_stock

        FROM
TABLE( STOCK_LEVEL( :w_id, :d_id,
:threshold ) ) AS T

        WITH CS;
*/

   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 SQL call");
#endif /* DEBUGIT */

   if ( sqlca.sqlcode != 0 )
   {
      DLCHK( retry_tran ) ;

      sqlerror( STOCKLEV_SQL ,
"STK" , __FILE__, __LINE__ , &sqlca);
      stocklev->s_transtatus =
FATAL_SQLERROR ;
      clientRc = FATAL_SQLERROR ;

      stk_debug( stocklev, in_stocklev,
"STK failed" ) ;

      EXEC SQL ROLLBACK WORK ;

      if ( sqlca.sqlcode != 0 )
      {
         sqlerror( STOCKLEV_SQL,
"ROLLBACK FAILED", __FILE__,
__LINE__, &sqlca ) ;
      }
   }

   return ( clientRc ) ;
}
```

Src.Common/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 - 2005

## 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.Common

#



!include
$(TPCC_ROOT)/Makefile.config



#
################################
################################
##########

# Preprocessor, Compiler and LInker
Flags

#
################################
################################
##########



BND_OPTS =GRANT PUBLIC \

      MESSAGES $*.bnd.msg

PRP_OPTS =BINDFILE \

      OPTLEVEL 1 \

      ISOLATION RR \

      MESSAGES $*.prep.msg \

      LEVEL $(TPCC_VERSION) \

      NOLINEMACRO



INCLUDES = -
I$(TPCC_SQLLIB)$(SLASH)include -
I$(TPCC_ROOT)$(SLASH)include



CFLAGS =  $(CFLAGS_OS)
$(CFLAGS_DEBUG) $(INCLUDES) \

      -DSQLA_NOLINES -
D$(DB2EDITION) -D$(DB2VERSION)
\

      -D$(TPCC_SPTYPE)



UTIL_OBJ =   tpccmisc$(OBJEXT)
tpccdbg$(OBJEXT) tpccctx$(OBJEXT)



#
################################
################################
##########

# User Targets

#
################################
################################
##########



all:  connect $(UTIL_OBJ) disconnect



clean:

   - $(ERASE) *$(OBJEXT) *.bnd
*.msg tpccctx.c



#
################################
################################
##########

# Helper Targets

#
################################
################################
##########



connect:

   - db2 connect to $(TPCC_DBNAME)



disconnect:

   - db2 connect reset

   - db2 terminate



rebind:

   db2 bind tpccctx.bnd $(BND_OPTS)



#
################################
################################
##########

# Build Rules

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

```
.SUFFIXES:

.SUFFIXES: $(OBJEXT) .c .sqc


.sqc.c:

   @echo "Prepping $*.sqc"

   -db2 prep $*.sqc $(PRP_OPTS)

   @echo "Binding $*.bnd"

   db2 bind $*.bnd $(BND_OPTS)


#
################################
################################
##########

# Dependencies

#
################################
################################
##########


# Source

tpccdbg$(OBJEXT): tpccdbg.c

tpccctx$(OBJEXT):  tpccctx.c

tpccmisc$(OBJEXT):  tpccmisc.c


# Headers

tpccdbg.c:
   $(TPCC_ROOT)/include/db2tpcc.h
```

Src.Common/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 - 2005

** 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 <string.h>

#include <sqlutil.h>

#include "db2tpcc.h"

#include "tpccdbg.h"


int connect_to_TM(char *in_dbname);

int connect_to_TM_auth(char
*in_dbname, char *in_username, char
*in_password);

int disconnect_from_TM(void);

int create_context();

int destroy_context();

int attach_context(void*);

int detach_context(void*);

int get_context(void**);


int connect_to_TM(char *in_dbname)

{

  return
connect_to_TM_auth(in_dbname, "",
"");

}


int connect_to_TM_auth(char
*in_dbname, char *in_username, char
*in_password)

{

   SQL_STRUCTURE sqlca  sqlca;

   int ConnectSQLCODE = 0;


   EXEC SQL BEGIN DECLARE
SECTION;

   char dbname[9];

   char username[129];

   char password[15];

   EXEC SQL END DECLARE
SECTION;


   SQLCODE = create_context();

   if (SQLCODE != 0) { return
SQLCODE; }


   /* Copy 9 characters - 8 for dbname,
1 for NULL */

   strncpy(dbname,in_dbname,9);

   if (strcmp(in_username,"") == 0)

   {

      EXEC SQL CONNECT
TO :dbname IN SHARE MODE;

   } else {

strncpy(username,in_username,128);

      strncpy(password,in_password,14);

      EXEC SQL CONNECT
TO :dbname IN SHARE MODE
USER :username USING :password;

   }


   ConnectSQLCODE = SQLCODE;

   if (ConnectSQLCODE != 0)

   {

      sqlerror( CLIENT_SQL,
"CONNECT", __FILE__, __LINE__,
&sqlca);


      SQLCODE = destroy_context();

      if (SQLCODE != 0) { return
SQLCODE; }
```

```
    return ConnectSQLCODE;

  }


  return 0;

}


int disconnect_from_TM(void)

{

  SQL_STRUCTURE sqlca  sqlca;

  int DisconnectSQLCODE = 0;


  EXEC SQL CONNECT RESET;


  DisconnectSQLCODE = SQLCODE;

  if (DisconnectSQLCODE != 0) {

    sqlerror( CLIENT_SQL,
"DISCONNECT", __FILE__, __LINE__,
&sqlca);

  }


  SQLCODE = destroy_context();

  if (SQLCODE != 0) { return
SQLCODE; }


  if (DisconnectSQLCODE) {

    return DisconnectSQLCODE;

  }

  return 0;

}


int create_context(void)

{

  SQL_STRUCTURE sqlca sqlca;

  void *ctx;


sqleSetTypeCtx(SQL_CTX_MULTI_M
ANUAL);

  sqleBeginCtx(&ctx,
SQL_CTX_BEGIN_ALL, NULL,
&sqlca);


  if (SQLCODE != 0) {

    sqlerror( CLIENT_SQL, "CREATE",
__FILE__, __LINE__, &sqlca);

    return SQLCODE;

  }


  return 0;

}


int attach_context(void *ctx)

{

  SQL_STRUCTURE sqlca sqlca;


  sqleAttachToCtx(ctx, NULL, &sqlca);


  if (SQLCODE != 0) {

    sqlerror( CLIENT_SQL, "ATTACH",
__FILE__, __LINE__, &sqlca);

    return SQLCODE;

  }


  return 0;

}


int detach_context(void *ctx)

{

  SQL_STRUCTURE sqlca sqlca;


  sqleDetachFromCtx(ctx, NULL,
&sqlca);


  if (SQLCODE != 0) {

    sqlerror( CLIENT_SQL, "DETACH",
__FILE__, __LINE__, &sqlca);

    return SQLCODE;

  }


  return 0;

}


int destroy_context(void)

{

  SQL_STRUCTURE sqlca sqlca;

  void *ctx;


  SQLCODE = get_context(&ctx);

  if (SQLCODE) { return SQLCODE; }


  sqleEndCtx(&ctx,
SQL_CTX_END_ALL, NULL, &sqlca);


  if (SQLCODE != 0) {

    sqlerror( CLIENT_SQL,
"DESTROY", __FILE__, __LINE__,
&sqlca);

    return SQLCODE;

  }


  return 0;

}


int get_context(void **ctx)

{

  SQL_STRUCTURE sqlca sqlca;


  sqleGetCurrentCtx(ctx, NULL,
&sqlca);


  if (SQLCODE != 0) {

    sqlerror( CLIENT_SQL, "GETCTX",
__FILE__, __LINE__, &sqlca);

    return SQLCODE;

  }
```

```
   return 0;

}
```

Src.Common/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 - 2005

** 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, "C:\\temp");

      }

      strcat(debugPath, "\\");

   }

   debugInit = 1;

}


/*-----------------------------------------------
-------------*/

/*        sqlerror
*/

/*-----------------------------------------------
-------------*/

void sqlerror(int tranType, char *msg,
char *file, int line, 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;


    case 0:

        strcat(err_fn, "cli.err.out");

        strcpy(tranName, "CLIENT");

        break;


    default:

        return;


  }


  /* Generate Formatted Error
Message */

  sqlaintp(errStr, 512, 78, psqlca);


  if ((err_fp = fopen(err_fn, "a+")) ==
NULL)

  {

    return;

  }


  fprintf(err_fp, "-----------------------------
----------\n");

  fprintf(err_fp, "Transaction: %s
(%s)\n", tranName, msg);

  fprintf(err_fp, "FILE %s (%u)\n", file,
line);

  fprintf(err_fp, "SQLCODE %d ",
psqlca->sqlcode);

  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,"\n===============
=============================
=====\n");



  fprintf(debug_fp,"in_delivery_struct
{\n");

  fprintf(debug_fp,"\ts_W_ID        =
%d (%X)\n",

        in_delivery->s_W_ID,
in_delivery->s_W_ID);


fprintf(debug_fp,"\ts_O_CARRIER_ID
= %d (%X)\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   =
%d (%X)\n",

        delivery_ptr-
>s_transtatus,delivery_ptr-
>s_transtatus);

  fprintf(debug_fp,"\tdeadlocks      =
%d (%X)\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,"\n===============
============================
=====\n");



   fprintf(debug_fp,"in_neword_struct
{\n");



   fprintf(debug_fp,"\ts_C_ID         = %d
(%X)\n",

         in_neword->s_C_ID,
in_neword->s_C_ID);

   fprintf(debug_fp,"\ts_W_ID         =
%d (%X)\n",

         in_neword->s_W_ID,
in_neword->s_W_ID);

   fprintf(debug_fp,"\ts_D_ID         = %d
(%X)\n",

         in_neword->s_D_ID,
in_neword->s_D_ID);

   fprintf(debug_fp,"\ts_O_OL_CNT
= %d (%X)\n",

         in_neword->s_O_OL_CNT,
in_neword->s_O_OL_CNT);

   fprintf(debug_fp,"\ts_all_local    = %d
(%X)\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   =
%d (%X)\n",

//       in_neword->s_transtatus,
in_neword->s_transtatus);

// fprintf(debug_fp,"\tduplicate_items=
%d (%X)\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]
= %d (%X)\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] = %d (%X)\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]    = %d (%X)\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         = %d
(%X)\n",

         neword_ptr->s_O_ID,
neword_ptr->s_O_ID);

   fprintf(debug_fp,"\ts_O_OL_CNT
= %d (%X)\n",

         neword_ptr->s_O_OL_CNT,
neword_ptr->s_O_OL_CNT);

   fprintf(debug_fp,"\ts_O_ENTRY_D
= %lld (%llX)\n",

         neword_ptr-
>s_O_ENTRY_D_time, neword_ptr-
>s_O_ENTRY_D_time);

   fprintf(debug_fp,"\ts_total_amount =
%d\n",

         neword_ptr->s_total_amount);

   fprintf(debug_fp,"\ts_transtatus   =
%d (%X)\n",

         neword_ptr->s_transtatus,
neword_ptr->s_transtatus);

   fprintf(debug_fp,"\tdeadlocks      =
%d (%X)\n",

         neword_ptr->deadlocks,
neword_ptr->deadlocks);



// fprintf(debug_fp,"\ts_W_ID         =
%d (%X)\n",

//       neword_ptr->s_W_ID,
neword_ptr->s_W_ID);

// fprintf(debug_fp,"\ts_D_ID         = %d
(%X)\n",

//       neword_ptr->s_D_ID,
neword_ptr->s_D_ID);
```

```
// fprintf(debug_fp,"\ts_all_local    = %d
(%X)\n",

//        neword_ptr->s_all_local,
neword_ptr->s_all_local);

// fprintf(debug_fp,"\tduplicate_items=
%d (%X)\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]      = %d\n",

            j, neword_ptr-
>item[j].s_OL_AMOUNT);


fprintf(debug_fp,"\t\ts_S_QUANTITY[
%d]     = %d (%X)\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,"\n===============
==============================
=====\n");



   fprintf(debug_fp,"in_ordstat_struct
{\n");

   fprintf(debug_fp,"\ts_W_ID        =
%d (%X)\n",

         in_ordstat->s_W_ID, in_ordstat-
>s_W_ID);

   fprintf(debug_fp,"\ts_D_ID        = %d
(%X)\n",

         in_ordstat->s_D_ID, in_ordstat-
>s_D_ID);

   fprintf(debug_fp,"\ts_C_ID        = %d
(%X)\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        = %d
(%X)\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         = %d
(%X)\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
= %d (%X)\n",

          ordstat_ptr->s_O_CARRIER_ID,
ordstat_ptr->s_O_CARRIER_ID);

   fprintf(debug_fp,"\ts_ol_cnt       = %d
(%X)\n",

          ordstat_ptr->s_ol_cnt,
ordstat_ptr->s_ol_cnt);

   fprintf(debug_fp,"\ts_transtatus   =
%d (%X)\n",

          ordstat_ptr->s_transtatus,
ordstat_ptr->s_transtatus);

   fprintf(debug_fp,"\tdeadlocks      =
%d (%X)\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] = %d (%X)\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]
= %d (%X)\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]    = %d (%X)\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,"\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        = %d
(%X)\n",

        in_payment->s_C_ID,
in_payment->s_C_ID);

  fprintf(debug_fp,"\ts_W_ID        =
%d (%X)\n",

        in_payment->s_W_ID,
in_payment->s_W_ID);

  fprintf(debug_fp,"\ts_D_ID        = %d
(%X)\n",

        in_payment->s_D_ID,
in_payment->s_D_ID);

  fprintf(debug_fp,"\ts_C_D_ID      =
%d (%X)\n",

        in_payment->s_C_D_ID,
in_payment->s_C_D_ID);

  fprintf(debug_fp,"\ts_C_W_ID      =
%d (%X)\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        = %d
(%X)\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  =
%d (%X)\n",

        payment_ptr-
>s_transtatus,payment_ptr-
>s_transtatus);

  fprintf(debug_fp,"\tdeadlocks     =
%d (%X)\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,"\n===============
==============================
=====\n");



   fprintf(debug_fp,"in_stocklev_struct
{\n");

   fprintf(debug_fp,"\ts_W_ID        =
%d (%X)\n",

            in_stocklev->s_W_ID,
in_stocklev->s_W_ID);

   fprintf(debug_fp,"\ts_D_ID        = %d
(%X)\n",

            in_stocklev->s_D_ID,
in_stocklev->s_D_ID);

   fprintf(debug_fp,"\ts_threshold    =
%d (%X)\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   =
%d (%X)\n",

            stocklev->s_transtatus,
stocklev->s_transtatus);

   fprintf(debug_fp,"\tdeadlocks      =
%d (%X)\n",

            stocklev->deadlocks, stocklev-
>deadlocks);

   fprintf(debug_fp,"\ts_low_stock    =
%d (%X)\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);

}
```

include/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 - 2005

** 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>

typedef __int16 int16_t;

typedef __int32 int32_t;

typedef __int64 int64_t;


#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 CLIENT_SQL   0

#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


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

/* DB2 Connect/Disconnect & Thread
Context Wrappers                    */

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


#ifdef __cplusplus

extern "C" {

#endif


extern int connect_to_TM(char*);

extern int connect_to_TM_auth(char*,
char*, char*);

extern int disconnect_from_TM(void);


extern int create_context(void);

extern int destroy_context(void);

extern int get_context(void**);

extern int attach_context(void*);

extern int detach_context(void*);


#ifdef __cplusplus

}

#endif


#endif // __DB2TPCC_H
```

include/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 - 2005

** 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 <string.h>

#include <time.h>


#include "sqlenv.h"

#define daricall __stdcall


#include "sqlca.h"

#include "sqlcodes.h"
```

```
#ifdef SWAP_ENDIAN

#define SWAP_BYTE(Var)
SwapEndian((void*)&Var, sizeof(Var))


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

  FUNCTION: SwapEndian

  PURPOSE: Swap the byte order of a
structure

  EXAMPLE: int I=0x12345678;
SWAP_BYTE(I); I => 0x78563412;

  IMPLEMENTATION: Fold Addr in
half, swap header & tail by XOR op

          e.g.: *a = 0x12 [ Addr + 0];

                *b = 0x78 [ Add + 4 - 0 - 1
= Addr+3];

                *a ^= *b;         // sets *a to
0x6A

                *b ^= *a;         // sets *b to
0x12

                *a ^= *b;         // sets *a to
0x78


           Now *a => 0x78 && *b =>
0x12

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


void SwapEndian(void *Addr, int nb)

{

  int i;

  for (i=0; i<nb/2; i++)

  {

    char *a = (char*)Addr+i;

    char *b = (char*)Addr+(nb-i-1);


    *a ^= *b;

    *b ^= *a;

    *a ^= *b;

  }

}

#endif   //SWAP_ENDIAN


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

/* 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
```

_include/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 - 2005

** 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


#ifdef __cplusplus

extern "C" {

#endif


extern void sqlerror (int tranType, char
*msg, char *file, int line,

                SQL_STRUCTURE
sqlca *psqlca);


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);


#ifdef __cplusplus

}

#endif

#endif // __TPCCDBG_H
```

_tpccenv.bat

```
@REM
*****************************************
****************************

@REM  Licensed Materials - Property
of IBM

@REM

@REM  Governed under the terms of
the International

@REM  License Agreement for Non-
Warranted Sample Code.

@REM

@REM  (C) COPYRIGHT International
Business Machines Corp. 1996 - 2005

@REM  All Rights Reserved.

@REM

@REM  US Government Users
Restricted Rights - Use, duplication or

@REM  disclosure restricted by GSA
ADP Schedule Contract with IBM Corp.

@REM
*****************************************
*****************************

@REM

@REM tpccenv.bat - Windows
Environment Setup

@REM



@REM The Kit Version

set TPCC_VERSION=CK041012



@REM The DB2 Instance Name (for
DB2)

set DB2INSTANCE=%USERNAME%


@REM The OS being used (i.e.
"UNIX", "WINDOWS")

set PLATFORM=WINDOWS

@REM The type of make command
and slash used by the OS

@REM (i.e. UNIX - "/", WINDOWS - "\")

@REM These are referenced all over
the kit.

set SLASH=\

set MAKE=nmake


set TPCC_SPTYPE=SPGENERAL


set DB2VERSION=v8


@REM The schema name is typically
the SQL autorization ID (or username).

@REM This is required for runstats
and EEE.

set TPCC_SCHEMA=%USERNAME%


@REM DB2 EE/EEE Configuration

set DB2EDITION=EE

set DB2NODE=0

set DB2NODES=1


set HOME=C:\home\tpcc

set TPCC_DBNAME=TPCC

set TPCC_ROOT=%HOME%\tpc-
c.ibm

set
TPCC_SQLLIB=C:\Progra~1\IBM\sqlli
b

set TPCC_RUNDATA=%HOME%\tpc-
c.ibm\tpccdata


set TPCC_DEBUGDIR=c:\temp


@REM Specifies where stored
procedures should be placed and if
they should

@REM be fenced.

set
TPCC_SPDIR=%TPCC_SQLLIB%\fun
ction
```

```
set TPCC_FENCED=NO
```

Transaction Code

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 - 2005
## All Rights Reserved.
##
## US Government Users Restricted
Rights - Use, duplication or
## disclosure restricted by GSA ADP
Schedule Contract with IBM Corp.
################################
################################
###############

#
# Makefile.config - Linux 32/64-bit
#

# Make Configuration
MAKE=make

# Compiler Configuration.
# CFLAGS_DEBUG may be set to "-g",
"-DDEBUGIT" "-g -DDEBUGIT" or left
blank
CC=cc
CFLAGS_OS=-m64 -DSQLUNIX -
DSQLLinux -O3 -fpic
#CFLAGS_OS=-DDEBUGIT -
DSQLUNIX -DSQLLinux -O3 -fpic
CFLAGS_OUT=-o
CFLAGS_DEBUG=

# Linker Configuration
LD_EXEC=cc
LD_STORP=ld
LDFLAGS_EXEC=-m64
LDFLAGS_STORP=-shared -m
elf64ppc
LDFLAGS_LIB=-
L$(TPCC_SQLLIB)/lib -ldb2
#LDFLAGS_LIB=-Xlinker -m -Xlinker
elf64ppc -L$(TPCC_SQLLIB)/lib -ldb2
LDFLAGS_OUT=-o

# Library Configuration
AR=ar
ARFLAGS=-r -v
ARFLAGS_LIB=
ARFLAGS_OUT=

# OS Commands
ERASE=rm -f
ERASEDIR=$(ERASE) -R
MOVE=mv
COPY=cp

# OS File Extensions & Path
Separators
OBJEXT=.o
LIBEXT=.a
BINEXT=
SLASH=/
CMDSEP=;
```

Src.Common/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 - 2005
## 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.Common
#

include
$(TPCC_ROOT)/Makefile.config

#
################################
################################
##########
# Preprocessor, Compiler and LInker
Flags
#
################################
################################
##########

BND_OPTS =GRANT PUBLIC \
    MESSAGES $*.bnd.msg
PRP_OPTS =BINDFILE \
    OPTLEVEL 1 \
    ISOLATION RR \
    MESSAGES $*.prep.msg \
    LEVEL $(TPCC_VERSION) \
    NOLINEMACRO

INCLUDE =-
I$(TPCC_SQLLIB)/include -
I$(TPCC_ROOT)/include

CFLAGS =  $(CFLAGS_OS)
$(CFLAGS_DEBUG) $(INCLUDE) \
    -DSQLA_NOLINES -
D$(DB2EDITION) -D$(DB2VERSION)
\
    -D$(TPCC_SPTYPE)

UTIL_OBJ =   tpccmisc$(OBJEXT)
tpccdbg$(OBJEXT) tpccctx$(OBJEXT)

#
################################
################################
##########
# User Targets
#
################################
################################
##########

all:  connect $(UTIL_OBJ) disconnect

clean:
  - $(ERASE) *$(OBJEXT) *.bnd
*.msg tpccctx.c

#
################################
################################
##########
# Helper Targets
#
################################
################################
##########

connect:
  - db2 connect to $(TPCC_DBNAME)

disconnect:
  - db2 connect reset
  - db2 terminate

rebind:  connect
  db2 bind tpccctx.bnd $(BND_OPTS)

#
################################
################################
##########
# Build Rules
#
################################
################################
##########

.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc

.sqc.c:
  @echo "Prepping $*.sqc"
  -db2 prep $*.sqc $(PRP_OPTS)
  @echo "Binding $*.bnd"
  db2 bind $*.bnd $(BND_OPTS)

#
################################
################################
##########
# Dependencies
#
################################
```

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

# Source
tpccdbg$(OBJEXT): tpccdbg.c
tpccctx$(OBJEXT):  tpccctx.c
tpccmisc$(OBJEXT):  tpccmisc.c

# Headers
tpccdbg.c:
   $(TPCC_ROOT)/include/db2tpcc.h
```

Src.Common/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 - 2005
** 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 <string.h>
#include <sqlutil.h>
#include "db2tpcc.h"
#include "tpccdbg.h"

int connect_to_TM(char *in_dbname);
int connect_to_TM_auth(char
*in_dbname, char *in_username, char
*in_password);
int disconnect_from_TM(void);

int connect_to_TM(char *in_dbname)
{
  return
connect_to_TM_auth(in_dbname, "",
"");
}

int connect_to_TM_auth(char
*in_dbname, char *in_username, char
*in_password)
{
   SQL_STRUCTURE sqlca  sqlca;
   int ConnectSQLCODE = 0;

   EXEC SQL BEGIN DECLARE
SECTION;
   char dbname[9];
   char username[129];
   char password[15];
   EXEC SQL END DECLARE
SECTION;

   /* Copy 9 characters - 8 for dbname,
1 for NULL */
   strncpy(dbname,in_dbname,9);
   if (strcmp(in_username,"") == 0)
   {
      EXEC SQL CONNECT
TO :dbname IN SHARE MODE;
   } else {

strncpy(username,in_username,128);
      strncpy(password,in_password,14);
      EXEC SQL CONNECT
TO :dbname IN SHARE MODE
USER :username USING :password;
   }

   ConnectSQLCODE = SQLCODE;
   if (ConnectSQLCODE != 0)
   {
      sqlerror( CLIENT_SQL,
"CONNECT", __FILE__, __LINE__,
&sqlca);

      return ConnectSQLCODE;
   }

   return 0;
}

int disconnect_from_TM(void)
{
   SQL_STRUCTURE sqlca  sqlca;
   int DisconnectSQLCODE = 0;

   EXEC SQL CONNECT RESET;

   DisconnectSQLCODE = SQLCODE;
   if (DisconnectSQLCODE != 0) {
      sqlerror( CLIENT_SQL,
"DISCONNECT", __FILE__, __LINE__,
&sqlca);
   }

   if (DisconnectSQLCODE) {
      return DisconnectSQLCODE;
   }
   return 0;
}
```

Src.Common/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 - 2005
** 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,
char *file, int line, 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;

      case 0:
         strcat(err_fn, "cli.err.out");
         strcpy(tranName, "CLIENT");
         break;

      default:
         return;

   }

   /* Generate Formatted Error
Message */
   sqlaintp(errStr, 512, 78, psqlca);

   if ((err_fp = fopen(err_fn, "a+")) ==
NULL)
   {
      return;
   }

   fprintf(err_fp, "----------------------------
----------\n");
   fprintf(err_fp, "Transaction: %s
(%s)\n", tranName, msg);
   fprintf(err_fp, "FILE %s (%u)\n", file,
line);
   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        =
%d (%X)\n",
         in_delivery->s_W_ID,
in_delivery->s_W_ID);

fprintf(debug_fp,"\ts_O_CARRIER_ID
= %d (%X)\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   =
%d (%X)\n",
         delivery_ptr-
>s_transtatus,delivery_ptr-
>s_transtatus);
   fprintf(debug_fp,"\tdeadlocks      =
%d (%X)\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         = %d
(%X)\n",
         in_neword->s_C_ID,
in_neword->s_C_ID);
   fprintf(debug_fp,"\ts_W_ID         =
%d (%X)\n",
         in_neword->s_W_ID,
in_neword->s_W_ID);
   fprintf(debug_fp,"\ts_D_ID         = %d
(%X)\n",
         in_neword->s_D_ID,
in_neword->s_D_ID);
   fprintf(debug_fp,"\ts_O_OL_CNT
= %d (%X)\n",
         in_neword->s_O_OL_CNT,
in_neword->s_O_OL_CNT);
   fprintf(debug_fp,"\ts_all_local    = %d
(%X)\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   =
%d (%X)\n",
//       in_neword->s_transtatus,
in_neword->s_transtatus);
// fprintf(debug_fp,"\tduplicate_items=
%d (%X)\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]
= %d (%X)\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] = %d (%X)\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]    = %d (%X)\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         = %d
(%X)\n",
         neword_ptr->s_O_ID,
neword_ptr->s_O_ID);
   fprintf(debug_fp,"\ts_O_OL_CNT
= %d (%X)\n",
         neword_ptr->s_O_OL_CNT,
neword_ptr->s_O_OL_CNT);
   fprintf(debug_fp,"\ts_O_ENTRY_D
= %lld (%llX)\n",
         neword_ptr-
>s_O_ENTRY_D_time, neword_ptr-
>s_O_ENTRY_D_time);
   fprintf(debug_fp,"\ts_total_amount =
%d\n",
         neword_ptr->s_total_amount);
   fprintf(debug_fp,"\ts_transtatus   =
%d (%X)\n",
         neword_ptr->s_transtatus,
neword_ptr->s_transtatus);
   fprintf(debug_fp,"\tdeadlocks      =
%d (%X)\n",
         neword_ptr->deadlocks,
neword_ptr->deadlocks);

// fprintf(debug_fp,"\ts_W_ID         =
%d (%X)\n",
//       neword_ptr->s_W_ID,
neword_ptr->s_W_ID);
// fprintf(debug_fp,"\ts_D_ID         = %d
(%X)\n",
//       neword_ptr->s_D_ID,
neword_ptr->s_D_ID);
// fprintf(debug_fp,"\ts_all_local    = %d
(%X)\n",
//       neword_ptr->s_all_local,
neword_ptr->s_all_local);
// fprintf(debug_fp,"\tduplicate_items=
%d (%X)\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]      = %d\n",
             j, neword_ptr-
>item[j].s_OL_AMOUNT);

fprintf(debug_fp,"\t\ts_S_QUANTITY[
%d]     = %d (%X)\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         =
%d (%X)\n",
             in_ordstat->s_W_ID, in_ordstat-
>s_W_ID);
   fprintf(debug_fp,"\ts_D_ID         = %d
(%X)\n",
             in_ordstat->s_D_ID, in_ordstat-
>s_D_ID);
   fprintf(debug_fp,"\ts_C_ID         = %d
(%X)\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         = %d
(%X)\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         = %d
(%X)\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
= %d (%X)\n",
             ordstat_ptr->s_O_CARRIER_ID,
ordstat_ptr->s_O_CARRIER_ID);
   fprintf(debug_fp,"\ts_ol_cnt       = %d
(%X)\n",
             ordstat_ptr->s_ol_cnt,
ordstat_ptr->s_ol_cnt);
   fprintf(debug_fp,"\ts_transtatus   =
%d (%X)\n",
             ordstat_ptr->s_transtatus,
ordstat_ptr->s_transtatus);
   fprintf(debug_fp,"\tdeadlocks      =
%d (%X)\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] = %d (%X)\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]
= %d (%X)\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]    = %d (%X)\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        = %d
(%X)\n",
        in_payment->s_C_ID,
in_payment->s_C_ID);
  fprintf(debug_fp,"\ts_W_ID        =
%d (%X)\n",
        in_payment->s_W_ID,
in_payment->s_W_ID);
  fprintf(debug_fp,"\ts_D_ID        = %d
(%X)\n",
        in_payment->s_D_ID,
in_payment->s_D_ID);
  fprintf(debug_fp,"\ts_C_D_ID      =
%d (%X)\n",
        in_payment->s_C_D_ID,
in_payment->s_C_D_ID);
  fprintf(debug_fp,"\ts_C_W_ID      =
%d (%X)\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        = %d
(%X)\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  =
%d (%X)\n",
        payment_ptr-
>s_transtatus,payment_ptr-
>s_transtatus);
  fprintf(debug_fp,"\tdeadlocks     =
%d (%X)\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        =
%d (%X)\n",
        in_stocklev->s_W_ID,
in_stocklev->s_W_ID);
  fprintf(debug_fp,"\ts_D_ID        = %d
(%X)\n",
        in_stocklev->s_D_ID,
in_stocklev->s_D_ID);
  fprintf(debug_fp,"\ts_threshold    =
%d (%X)\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   =
%d (%X)\n",
        stocklev->s_transtatus,
stocklev->s_transtatus);
  fprintf(debug_fp,"\tdeadlocks      =
%d (%X)\n",
        stocklev->deadlocks, stocklev-
>deadlocks);
  fprintf(debug_fp,"\ts_low_stock    =
%d (%X)\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);
}
```

Src.Common/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 - 2005
** 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 <stdlib.h>
#include <sys/types.h>
#include <sys/time.h>

double current_time_ms(void);
double current_time(void);


/* Current time in SECONDS,
precision SECONDS */
double current_time(void)
{
  /* use time() to get seconds */
  return(time(NULL));
}

/* Current time in SECONDS,
precision MILLISECONDS */
double current_time_ms(void)
{
   /* gettimeofday() returns seconds
and microseconds */
   /* convert to fractional seconds */
  struct timeval t;
  gettimeofday(&t,NULL);
  return (t.tv_sec +
(double)t.tv_usec/(1000*1000));
}
```

Src.Srv/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 - 2005
## 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

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 =
  $(TPCC_ROOT)/Src.Common/tpcc
misc$(OBJEXT) \

  $(TPCC_ROOT)/Src.Common/tpccd
bg$(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
   - $(ERASE) *.bnd *.msg *.out
*$(OBJEXT) $(EXE) tpcc_all_sql.c
   - $(ERASE) TPCC_ALL.*.plan

#
################################
################################
##########
# Helper Targets
#
################################
################################
##########

catalog:uncatalog
   - perl
$(TPCC_ROOT)$(SLASH)utils$(SLAS
H)genproc.pl $(STORED_PROCS)
   - db2 -tvf cat-proc.ddl +o -z cat-
proc.out
   - db2 -td$$ -vf cat-func.ddl +o -z cat-
func.out

uncatalog:
   - perl
$(TPCC_ROOT)$(SLASH)utils$(SLAS
H)genproc.pl $(STORED_PROCS)
   - db2 -td$$ -vf uncat-func.ddl +o -z
uncat-func.out
   - db2 -tvf uncat-proc.ddl +o -z uncat-
proc.out

explain:
   - perl
$(TPCC_ROOT)$(SLASH)utils$(SLAS
H)fixup_explain.pl
   - db2 -tvf
$(TPCC_ROOT)$(SLASH)utils$(SLAS
H)EXPLAIN.DDL +o -z EXPLAIN.out

unexplain:
   - db2 -tvf
$(TPCC_ROOT)$(SLASH)utils$(SLAS
H)UNEXPLAIN.DDL +o -z
UNEXPLAIN.out

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 catalog
   db2 bind tpcc_all_sql.bnd
$(BND_OPTS) QUERYOPT 7

#
################################
################################
##########
# 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 "Prepping $*.sqc"
   -db2 prep $*.sqc $(PRP_OPTS)
ISOLATION RR
   @echo "Binding $*.bnd"
   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
```

_Src.Srv/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 - 2005
-- 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 ;

  /* Delete the order from new order
table */
```

```
  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
            )
  ;

  /* Update the order as delivered and
retrieve the customer id */

  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
             )
  ;

  /* Charge the customer */

  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 the order id to the caller (or
NULL) */

  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

READS 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;

  /* Retrieve the Customer information
*/

  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

READS 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;

  /* Retrieve the Customer information
*/

  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) ;

  /* Update District and retieve its data
*/

  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
      )
  ;

  /* Determine the C_ID */

  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 )
    )
  ;


  /* Update the middle customer */

  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
    )
  ;

  /* Update the warehouse */

  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
    )
  ;

  /* Finally insert into the warehouse */

  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
          )
  ;

  /* Done - return the collected data */

  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) ;


  /* Update District and retieve its data
*/

  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
      )
  ;

  /* Update the middle customer */

  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
      )
  ;

  /* Update the warehouse */

  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
      )
  ;

  /* Finally insert into the warehouse */

  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
          )
  ;

  /* Done - return the collected data */

  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
                    )

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
                )

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
$
```

Src.Srv/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;
```

Src.Srv/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 - 2005
** 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  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      storedProcRc ;
  int      inputItemArrayIndex ;

  char    stockDistrictInformationArray
[15][25];

  #define stockDistrictInformation
stockDistrictInformationArray[ inputIte
mArrayIndex ]

  // 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_QU
ANTITY

  // This output field becomes an input
field to order_line

  #define next_o_id neword->s_O_ID


  #define  item_name    neword-
>item[ inputItemArrayIndex ].s_I_NAM
E

  sqlint32    i_priceArray[ 15 ] ;

  #define  item_price
i_priceArray[ inputItemArrayIndex ]

  // Handle the generic/brand
distinction

  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 ]


  // Redirect hostvars to input structure

  #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

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                , 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_SUPPLY_W_ID
                                , I_QTY

                            FROM
Table( VALUES


(  SMALLINT( 1 )        , :id0  , :ol_quan
tity0  , :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
                                    )
                            ) AS
NEW_OL_ALL

                        WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
              )


     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
                          ,
OL_DELIVERY_D
                          , I_QTY
                          ,
TOTAL_PRICE
                          ,
OL_DIST_INFO
                          , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                      FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_2 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                , 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_SUPPLY_W_ID
                                , I_QTY

                            FROM
Table( VALUES


(  SMALLINT( 1 )        , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                    ,
(  SMALLINT( 2 )        , :id1  , :ol_quan
tity1  , :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
                                    )
                            ) AS
NEW_OL_ALL

                        WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
              )


     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
                       ,
OL_DELIVERY_D
                       , I_QTY
                       ,
TOTAL_PRICE
                       ,
OL_DIST_INFO
                       , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                  FROM DATA

            ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_3 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
            , 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_SUPPLY_W_ID
                          , I_QTY

                      FROM
Table( VALUES


( SMALLINT( 1 )      , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                  ,
( SMALLINT( 2 )      , :id1  , :ol_quan
tity1  , :supply_w_id1  )
                                  ,
( SMALLINT( 3 )      , :id2  , :ol_quan
tity2  , :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
                                  )
                              ) AS
NEW_OL_ALL

                      WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
            )


    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
                       , I_QTY
                       ,
TOTAL_PRICE
                       ,
OL_DIST_INFO
                       , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                  FROM DATA

            ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_4 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
            , 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_SUPPLY_W_ID
                          , I_QTY

                      FROM
Table( VALUES


( SMALLINT( 1 )      , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                  ,
( SMALLINT( 2 )      , :id1  , :ol_quan
tity1  , :supply_w_id1  )
                                  ,
( SMALLINT( 3 )      , :id2  , :ol_quan
tity2  , :supply_w_id2  )
                                  ,
( SMALLINT( 4 )      , :id3  , :ol_quan
tity3  , :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
                                )
                            ) AS
NEW_OL_ALL

                        WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
                )


     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
                         ,
OL_DELIVERY_D
                         , I_QTY
                         ,
TOTAL_PRICE
                         ,
OL_DIST_INFO
                         , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                    FROM DATA
```

```
                ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_5 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                  , 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_SUPPLY_W_ID
                                , I_QTY

                              FROM
Table( VALUES


(  SMALLINT( 1 )        , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                ,
(  SMALLINT( 2 )        , :id1  , :ol_quan
tity1  , :supply_w_id1  )
                                ,
(  SMALLINT( 3 )        , :id2  , :ol_quan
tity2  , :supply_w_id2  )
                                ,
(  SMALLINT( 4 )        , :id3  , :ol_quan
tity3  , :supply_w_id3  )
                                ,
(  SMALLINT( 5 )        , :id4  , :ol_quan
tity4  , :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
                                )
```

```
                            ) AS
NEW_OL_ALL

                        WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
                )


     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
                         ,
OL_DELIVERY_D
                         , I_QTY
                         ,
TOTAL_PRICE
                         ,
OL_DIST_INFO
                         , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                    FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_6 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                , 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_SUPPLY_W_ID
                                , I_QTY

                            FROM
Table( VALUES


(  SMALLINT( 1 )        , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                ,
(  SMALLINT( 2 )        , :id1  , :ol_quan
tity1  , :supply_w_id1  )
                                ,
(  SMALLINT( 3 )        , :id2  , :ol_quan
tity2  , :supply_w_id2  )
                                ,
(  SMALLINT( 4 )        , :id3  , :ol_quan
tity3  , :supply_w_id3  )
                                ,
(  SMALLINT( 5 )        , :id4  , :ol_quan
tity4  , :supply_w_id4  )
                                ,
(  SMALLINT( 6 )        , :id5  , :ol_quan
tity5  , :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
                                )
                            ) AS
NEW_OL_ALL

                        WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
                )


     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
                             ,
OL_DELIVERY_D
                             , I_QTY
                             ,
TOTAL_PRICE
                             ,
OL_DIST_INFO
                             , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                        FROM DATA

                    ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_7 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                , 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_SUPPLY_W_ID
                                , I_QTY

                            FROM
Table( VALUES


(  SMALLINT( 1 )        , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                ,
(  SMALLINT( 2 )        , :id1  , :ol_quan
tity1  , :supply_w_id1  )
                                ,
(  SMALLINT( 3 )        , :id2  , :ol_quan
tity2  , :supply_w_id2  )
                                ,
(  SMALLINT( 4 )        , :id3  , :ol_quan
tity3  , :supply_w_id3  )
                                ,
(  SMALLINT( 5 )        , :id4  , :ol_quan
tity4  , :supply_w_id4  )
                                ,
(  SMALLINT( 6 )        , :id5  , :ol_quan
tity5  , :supply_w_id5  )
                                ,
(  SMALLINT( 7 )        , :id6  , :ol_quan
tity6  , :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
                                )
                            ) AS
NEW_OL_ALL

                        WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
                )


     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
                          ,
OL_DELIVERY_D
                          , I_QTY
                          ,
TOTAL_PRICE
                          ,
OL_DIST_INFO
                          , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                      FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_8 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
              , 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_SUPPLY_W_ID
                              , I_QTY
                      FROM
Table( VALUES

( SMALLINT( 1 )       , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                      ,
( SMALLINT( 2 )       , :id1  , :ol_quan
tity1  , :supply_w_id1  )
                                      ,
( SMALLINT( 3 )       , :id2  , :ol_quan
tity2  , :supply_w_id2  )
                                      ,
( SMALLINT( 4 )       , :id3  , :ol_quan
tity3  , :supply_w_id3  )
                                      ,
( SMALLINT( 5 )       , :id4  , :ol_quan
tity4  , :supply_w_id4  )
                                      ,
( SMALLINT( 6 )       , :id5  , :ol_quan
tity5  , :supply_w_id5  )
                                      ,
( SMALLINT( 7 )       , :id6  , :ol_quan
tity6  , :supply_w_id6  )
                                      ,
( SMALLINT( 8 )       , :id7  , :ol_quan
tity7  , :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
                                      )
                              ) AS
NEW_OL_ALL

                      WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
              )


    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
                          ,
OL_DELIVERY_D
                          , I_QTY
                          ,
TOTAL_PRICE
                          ,
OL_DIST_INFO
                          , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                      FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_9 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
              , 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_SUPPLY_W_ID
                              , I_QTY
                      FROM
Table( VALUES
```

```
( SMALLINT( 1 )       , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                        ,
( SMALLINT( 2 )       , :id1  , :ol_quan
tity1  , :supply_w_id1  )
                                        ,
( SMALLINT( 3 )       , :id2  , :ol_quan
tity2  , :supply_w_id2  )
                                        ,
( SMALLINT( 4 )       , :id3  , :ol_quan
tity3  , :supply_w_id3  )
                                        ,
( SMALLINT( 5 )       , :id4  , :ol_quan
tity4  , :supply_w_id4  )
                                        ,
( SMALLINT( 6 )       , :id5  , :ol_quan
tity5  , :supply_w_id5  )
                                        ,
( SMALLINT( 7 )       , :id6  , :ol_quan
tity6  , :supply_w_id6  )
                                        ,
( SMALLINT( 8 )       , :id7  , :ol_quan
tity7  , :supply_w_id7  )
                                        ,
( SMALLINT( 9 )       , :id8  , :ol_quan
tity8  , :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
                                        )
                                  ) AS
NEW_OL_ALL

                        WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
                )


     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
                             ,
OL_DELIVERY_D
                             , I_QTY
                             ,
TOTAL_PRICE
                             ,
OL_DIST_INFO
                             , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                        FROM DATA

                  ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_10 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                  , 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_SUPPLY_W_ID
                                  , I_QTY

                              FROM
Table( VALUES
```

```
( SMALLINT( 1 )       , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                        ,
( SMALLINT( 2 )       , :id1  , :ol_quan
tity1  , :supply_w_id1  )
                                        ,
( SMALLINT( 3 )       , :id2  , :ol_quan
tity2  , :supply_w_id2  )
                                        ,
( SMALLINT( 4 )       , :id3  , :ol_quan
tity3  , :supply_w_id3  )
                                        ,
( SMALLINT( 5 )       , :id4  , :ol_quan
tity4  , :supply_w_id4  )
                                        ,
( SMALLINT( 6 )       , :id5  , :ol_quan
tity5  , :supply_w_id5  )
                                        ,
( SMALLINT( 7 )       , :id6  , :ol_quan
tity6  , :supply_w_id6  )
                                        ,
( SMALLINT( 8 )       , :id7  , :ol_quan
tity7  , :supply_w_id7  )
                                        ,
( SMALLINT( 9 )       , :id8  , :ol_quan
tity8  , :supply_w_id8  )
                                        ,
( SMALLINT( 10 )      , :id9  , :ol_qua
ntity9  , :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
                                        )
                                  ) AS
NEW_OL_ALL

                        WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
                )


     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
                          ,
OL_DELIVERY_D
                          , I_QTY
                          ,
TOTAL_PRICE
                          ,
OL_DIST_INFO
                          , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                      FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_11 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
              , 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_SUPPLY_W_ID
                                , I_QTY

                            FROM
Table( VALUES

( SMALLINT( 1 )       , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                        ,
( SMALLINT( 2 )       , :id1  , :ol_quan
tity1  , :supply_w_id1  )
                                        ,
( SMALLINT( 3 )       , :id2  , :ol_quan
tity2  , :supply_w_id2  )
                                        ,
( SMALLINT( 4 )       , :id3  , :ol_quan
tity3  , :supply_w_id3  )
                                        ,
( SMALLINT( 5 )       , :id4  , :ol_quan
tity4  , :supply_w_id4  )
                                        ,
( SMALLINT( 6 )       , :id5  , :ol_quan
tity5  , :supply_w_id5  )
                                        ,
( SMALLINT( 7 )       , :id6  , :ol_quan
tity6  , :supply_w_id6  )
                                        ,
( SMALLINT( 8 )       , :id7  , :ol_quan
tity7  , :supply_w_id7  )
                                        ,
( SMALLINT( 9 )       , :id8  , :ol_quan
tity8  , :supply_w_id8  )
                                        ,
( SMALLINT( 10 )      , :id9  , :ol_qua
ntity9  , :supply_w_id9   )
                                        ,
( SMALLINT( 11 )      , :id10 , :ol_qua
ntity10 , :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
                                        )
                                ) AS
NEW_OL_ALL

                        WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
                )


    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
                          ,
OL_DELIVERY_D
                          , I_QTY
                          ,
TOTAL_PRICE
                          ,
OL_DIST_INFO
                          , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                      FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_12 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
              , 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_SUPPLY_W_ID
```

```
                                , I_QTY
                                FROM
Table( VALUES

( SMALLINT( 1 )       , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                ,
( SMALLINT( 2 )       , :id1  , :ol_quan
tity1  , :supply_w_id1  )
                                ,
( SMALLINT( 3 )       , :id2  , :ol_quan
tity2  , :supply_w_id2  )
                                ,
( SMALLINT( 4 )       , :id3  , :ol_quan
tity3  , :supply_w_id3  )
                                ,
( SMALLINT( 5 )       , :id4  , :ol_quan
tity4  , :supply_w_id4  )
                                ,
( SMALLINT( 6 )       , :id5  , :ol_quan
tity5  , :supply_w_id5  )
                                ,
( SMALLINT( 7 )       , :id6  , :ol_quan
tity6  , :supply_w_id6  )
                                ,
( SMALLINT( 8 )       , :id7  , :ol_quan
tity7  , :supply_w_id7  )
                                ,
( SMALLINT( 9 )       , :id8  , :ol_quan
tity8  , :supply_w_id8  )
                                ,
( SMALLINT( 10 )       , :id9  , :ol_qua
ntity9  , :supply_w_id9   )
                                ,
( SMALLINT( 11 )       , :id10 , :ol_qua
ntity10 , :supply_w_id10  )
                                ,
( SMALLINT( 12 )       , :id11 , :ol_qua
ntity11 , :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
                                )
                                ) AS
NEW_OL_ALL

                        WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
                )


     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
                          ,
OL_DELIVERY_D
                          , I_QTY
                          ,
TOTAL_PRICE
                          ,
OL_DIST_INFO
                          , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                      FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_13 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
              , 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_SUPPLY_W_ID
                              , I_QTY

                              FROM
Table( VALUES

( SMALLINT( 1 )       , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                ,
( SMALLINT( 2 )       , :id1  , :ol_quan
tity1  , :supply_w_id1  )
                                ,
( SMALLINT( 3 )       , :id2  , :ol_quan
tity2  , :supply_w_id2  )
                                ,
( SMALLINT( 4 )       , :id3  , :ol_quan
tity3  , :supply_w_id3  )
                                ,
( SMALLINT( 5 )       , :id4  , :ol_quan
tity4  , :supply_w_id4  )
                                ,
( SMALLINT( 6 )       , :id5  , :ol_quan
tity5  , :supply_w_id5  )
                                ,
( SMALLINT( 7 )       , :id6  , :ol_quan
tity6  , :supply_w_id6  )
                                ,
( SMALLINT( 8 )       , :id7  , :ol_quan
tity7  , :supply_w_id7  )
                                ,
( SMALLINT( 9 )       , :id8  , :ol_quan
tity8  , :supply_w_id8  )
                                ,
( SMALLINT( 10 )       , :id9  , :ol_qua
ntity9  , :supply_w_id9   )
                                ,
( SMALLINT( 11 )       , :id10 , :ol_qua
ntity10 , :supply_w_id10  )
                                ,
( SMALLINT( 12 )       , :id11 , :ol_qua
ntity11 , :supply_w_id11  )
                                ,
( SMALLINT( 13 )       , :id12 , :ol_qua
ntity12 , :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
                                )
```

```
                                ) AS
NEW_OL_ALL

                            WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
                )


     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
                               ,
OL_DELIVERY_D
                               , I_QTY
                               ,
TOTAL_PRICE
                               ,
OL_DIST_INFO
                               , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                          FROM DATA

                  ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_14 CURSOR FOR

     WITH DATA AS (  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 ) AS
TOTAL_PRICE
                  , 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_SUPPLY_W_ID
                                , I_QTY

                              FROM
Table( VALUES


(  SMALLINT( 1 )        , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                          ,
(  SMALLINT( 2 )        , :id1  , :ol_quan
tity1  , :supply_w_id1  )
                                          ,
(  SMALLINT( 3 )        , :id2  , :ol_quan
tity2  , :supply_w_id2  )
                                          ,
(  SMALLINT( 4 )        , :id3  , :ol_quan
tity3  , :supply_w_id3  )
                                          ,
(  SMALLINT( 5 )        , :id4  , :ol_quan
tity4  , :supply_w_id4  )
                                          ,
(  SMALLINT( 6 )        , :id5  , :ol_quan
tity5  , :supply_w_id5  )
                                          ,
(  SMALLINT( 7 )        , :id6  , :ol_quan
tity6  , :supply_w_id6  )
                                          ,
(  SMALLINT( 8 )        , :id7  , :ol_quan
tity7  , :supply_w_id7  )
                                          ,
(  SMALLINT( 9 )        , :id8  , :ol_quan
tity8  , :supply_w_id8  )
                                          ,
(  SMALLINT( 10 )       , :id9  , :ol_qua
ntity9  , :supply_w_id9   )
                                          ,
(  SMALLINT( 11 )       , :id10 , :ol_qua
ntity10 , :supply_w_id10  )
                                          ,
(  SMALLINT( 12 )       , :id11 , :ol_qua
ntity11 , :supply_w_id11  )
                                          ,
(  SMALLINT( 13 )       , :id12 , :ol_qua
ntity12 , :supply_w_id12  )
                                          ,
(  SMALLINT( 14 )       , :id13 , :ol_qua
ntity13 , :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
                                      )
                                ) AS
NEW_OL_ALL

                            WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
                )


     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
                               ,
OL_DELIVERY_D
                               , I_QTY
                               ,
TOTAL_PRICE
                               ,
OL_DIST_INFO
                               , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                          FROM DATA
```

```
                ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Remote_15 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                , 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_SUPPLY_W_ID
                                , I_QTY

                              FROM
Table( VALUES


(  SMALLINT( 1 )        , :id0  , :ol_quan
tity0  , :supply_w_id0  )
                                ,
(  SMALLINT( 2 )        , :id1  , :ol_quan
tity1  , :supply_w_id1  )
                                ,
(  SMALLINT( 3 )        , :id2  , :ol_quan
tity2  , :supply_w_id2  )
                                ,
(  SMALLINT( 4 )        , :id3  , :ol_quan
tity3  , :supply_w_id3  )
                                ,
(  SMALLINT( 5 )        , :id4  , :ol_quan
tity4  , :supply_w_id4  )
                                ,
(  SMALLINT( 6 )        , :id5  , :ol_quan
tity5  , :supply_w_id5  )
                                ,
(  SMALLINT( 7 )        , :id6  , :ol_quan
tity6  , :supply_w_id6  )
                                ,
(  SMALLINT( 8 )        , :id7  , :ol_quan
tity7  , :supply_w_id7  )
                                ,
(  SMALLINT( 9 )        , :id8  , :ol_quan
tity8  , :supply_w_id8  )
                                ,
(  SMALLINT( 10 )       , :id9  , :ol_qua
ntity9  , :supply_w_id9   )
                                ,
(  SMALLINT( 11 )       , :id10 , :ol_qua
ntity10 , :supply_w_id10  )
                                ,
(  SMALLINT( 12 )       , :id11 , :ol_qua
ntity11 , :supply_w_id11  )
                                ,
(  SMALLINT( 13 )       , :id12 , :ol_qua
ntity12 , :supply_w_id12  )
                                ,
(  SMALLINT( 14 )       , :id13 , :ol_qua
ntity13 , :supply_w_id13  )
                                ,
(  SMALLINT( 15 )       , :id14 , :ol_qua
ntity14 , :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
                                      )
                                ) AS
NEW_OL_ALL

                          WHERE
NEW_OL_ALL.I_PRICE IS NOT NULL
                )


    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
                          ,
OL_DELIVERY_D
                          , I_QTY
                          ,
TOTAL_PRICE
                          ,
OL_DIST_INFO
                          , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                      FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Local_1 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                , 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
                                      )
```

```
                                ) AS
NEW_OL_LOCAL

                            WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
                )


     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
                               ,
OL_DELIVERY_D
                               , I_QTY
                               ,
TOTAL_PRICE
                               ,
OL_DIST_INFO
                               , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                          FROM DATA

                  ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Local_2 CURSOR FOR

     WITH DATA AS (  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 ) AS
TOTAL_PRICE
                  , 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
                                        )
                                ) AS
NEW_OL_LOCAL

                            WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
                )


     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
                               ,
OL_DELIVERY_D
                               , I_QTY
                               ,
TOTAL_PRICE
                               ,
OL_DIST_INFO
                               , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                          FROM DATA

                  ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Local_3 CURSOR FOR

     WITH DATA AS (  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 ) AS
TOTAL_PRICE
                  , 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
                                        )
                                ) AS
NEW_OL_LOCAL

                        WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
                )


     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
                             ,
OL_DELIVERY_D
                             , I_QTY
                             ,
TOTAL_PRICE
                             ,
OL_DIST_INFO
                             , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                        FROM DATA

                    ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Local_4 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                  , 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
                                        )
                                ) AS
NEW_OL_LOCAL

                        WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
                )


     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
                             ,
OL_DELIVERY_D
                             , I_QTY
                             ,
TOTAL_PRICE
                             ,
OL_DIST_INFO
                             , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                        FROM DATA

                    ) AS INS
  ;
```

```
   EXEC SQL DECLARE
ISOL_Local_5 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                , 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
                                )
                            ) AS
NEW_OL_LOCAL

                        WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
                )


     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
                            ,
OL_DELIVERY_D
                            , I_QTY
                            ,
TOTAL_PRICE
                            ,
OL_DIST_INFO
                            , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                        FROM DATA

                  ) AS INS
   ;

   EXEC SQL DECLARE
ISOL_Local_6 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                , OL_DIST_INFO
                , I_PRICE, I_NAME,
I_DATA, S_DATA, S_QUANTITY

                        FROM
( SELECT   :next_o_id as O_ID
                                , :w_id
AS W_ID
                                , :d_id
as D_ID
                                ,
OL_NUMBER
                                , I_ID
                                , I_QTY

                            FROM
Table( VALUES


(  SMALLINT( 1 )  , :id0  , :ol_quantity0
  )
                                ,
(  SMALLINT( 2 )  , :id1  , :ol_quantity1
  )
                                ,
(  SMALLINT( 3 )  , :id2  , :ol_quantity2
  )
                                ,
(  SMALLINT( 4 )  , :id3  , :ol_quantity3
  )
                                ,
(  SMALLINT( 5 )  , :id4  , :ol_quantity4
  )
                                ,
(  SMALLINT( 6 )  , :id5  , :ol_quantity5
  )

                                ) AS
X ( OL_NUMBER , I_ID  ,
I_QTY          )
                        ) AS
ITEMLIST

                        ,
TABLE( NEW_OL_LOCAL(   I_ID
                                ,
I_QTY
                                ,
W_ID
                                ,
O_ID
                                ,
D_ID
                                )
                            ) AS
NEW_OL_LOCAL

                        WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
                )


     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
                           ,
OL_DELIVERY_D
                           , I_QTY
                           ,
TOTAL_PRICE
                           ,
OL_DIST_INFO
                           , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                      FROM DATA

                  ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Local_7 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                , 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
                                )
                              ) AS
NEW_OL_LOCAL

                      WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
                )


    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
                           ,
OL_DELIVERY_D
                           , I_QTY
                           ,
TOTAL_PRICE
                           ,
OL_DIST_INFO
                           , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                      FROM DATA

                  ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Local_8 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                , 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
                                        )
                                ) AS
NEW_OL_LOCAL

                        WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
                )


     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
                              ,
OL_DELIVERY_D
                              , I_QTY
                              ,
TOTAL_PRICE
                              ,
OL_DIST_INFO
                              , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                        FROM DATA

                  ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Local_9 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                  , 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
                                        )
                                ) AS
NEW_OL_LOCAL

                        WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
                )


     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
                       ,
OL_DELIVERY_D
                       , I_QTY
                       ,
TOTAL_PRICE
                       ,
OL_DIST_INFO
                       , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                  FROM DATA

            ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Local_10 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
            , 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_quantity
9  )

                       ) AS
X ( OL_NUMBER , I_ID  ,
I_QTY        )
                     ) AS
ITEMLIST

                     ,
TABLE( NEW_OL_LOCAL(   I_ID
                       ,
I_QTY
                       ,
W_ID
                       ,
O_ID
                       ,
D_ID
                       )
                     ) AS
NEW_OL_LOCAL

                  WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
            )


    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
                       ,
OL_DELIVERY_D
                       , I_QTY
                       ,
TOTAL_PRICE
                       ,
OL_DIST_INFO
                       , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                  FROM DATA

            ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Local_11 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
            , 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_quantity
9 )
                                        ,
( SMALLINT( 11 ) , :id10 , :ol_quantity
10 )
                                        ) AS
X ( OL_NUMBER , I_ID  ,
I_QTY          )
                                  ) AS
ITEMLIST

                                  ,
TABLE( NEW_OL_LOCAL(   I_ID
                                        ,
I_QTY
                                        ,
W_ID
                                        ,
O_ID
                                        ,
D_ID
                                        )
                                  ) AS
NEW_OL_LOCAL

                            WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
                )


     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
                                   ,
OL_DELIVERY_D
                                   , I_QTY
                                   ,
TOTAL_PRICE
                                   ,
OL_DIST_INFO
                                   , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                            FROM DATA

                      ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Local_12 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
                  , 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_quantity
9 )
                                        ,
( SMALLINT( 11 ) , :id10 , :ol_quantity
10 )
                                        ,
( SMALLINT( 12 ) , :id11 , :ol_quantity
11 )

                                        ) AS
X ( OL_NUMBER , I_ID  ,
I_QTY          )
                                  ) AS
ITEMLIST

                                  ,
TABLE( NEW_OL_LOCAL(   I_ID
                                        ,
I_QTY
                                        ,
W_ID
                                        ,
O_ID
                                        ,
D_ID
                                        )
                                  ) AS
NEW_OL_LOCAL

                            WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
                )


     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
                           ,
OL_DELIVERY_D
                           , I_QTY
                           ,
TOTAL_PRICE
                           ,
OL_DIST_INFO
                           , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                      FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Local_13 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
              , 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_quantity
9  )
                                        ,
(  SMALLINT( 11 ) , :id10 , :ol_quantity
10 )
                                        ,
(  SMALLINT( 12 ) , :id11 , :ol_quantity
11 )
                                        ,
(  SMALLINT( 13 ) , :id12 , :ol_quantity
12 )

                                      ) AS
X ( OL_NUMBER , I_ID  ,
I_QTY         )
                                ) AS
ITEMLIST

                                ,
TABLE( NEW_OL_LOCAL(   I_ID
                                        ,
I_QTY
                                        ,
W_ID
                                        ,
O_ID
                                        ,
D_ID
                                        )
                                ) AS
NEW_OL_LOCAL

                        WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
              )


     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
                           ,
OL_DELIVERY_D
                           , I_QTY
                           ,
TOTAL_PRICE
                           ,
OL_DIST_INFO
                           , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                      FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Local_14 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
              , 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_quantity
9  )
                                ,
( SMALLINT( 11 ) , :id10 , :ol_quantity
10 )
                                ,
( SMALLINT( 12 ) , :id11 , :ol_quantity
11 )
                                ,
( SMALLINT( 13 ) , :id12 , :ol_quantity
12 )
                                ,
( SMALLINT( 14 ) , :id13 , :ol_quantity
13 )

                                ) AS
X ( OL_NUMBER , I_ID  ,
I_QTY          )
                            ) AS
ITEMLIST

                        ,
TABLE( NEW_OL_LOCAL(   I_ID
                                ,
I_QTY
                                ,
W_ID
                                ,
O_ID
                                ,
D_ID
                                )
                            ) AS
NEW_OL_LOCAL

                        WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
                )
```

```
    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
                             ,
OL_DELIVERY_D
                             , I_QTY
                             ,
TOTAL_PRICE
                             ,
OL_DIST_INFO
                             , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                        FROM DATA

                ) AS INS
  ;

  EXEC SQL DECLARE
ISOL_Local_15 CURSOR FOR

    WITH DATA AS (  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 ) AS
TOTAL_PRICE
              , 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_quantity
9  )
                                ,
( SMALLINT( 11 ) , :id10 , :ol_quantity
10 )
                                ,
( SMALLINT( 12 ) , :id11 , :ol_quantity
11 )
                                ,
( SMALLINT( 13 ) , :id12 , :ol_quantity
12 )
                                ,
( SMALLINT( 14 ) , :id13 , :ol_quantity
13 )
                                ,
( SMALLINT( 15 ) , :id14 , :ol_quantity
14 )

                                ) AS
X ( OL_NUMBER , I_ID  ,
I_QTY          )
                            ) AS
ITEMLIST

                        ,
TABLE( NEW_OL_LOCAL(   I_ID
```

```
                                        ,
I_QTY
                                        ,
W_ID
                                        ,
O_ID
                                        ,
D_ID
                                        )
                                ) AS
NEW_OL_LOCAL

                        WHERE
NEW_OL_LOCAL.I_PRICE IS NOT
NULL
                )


     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
                              ,
OL_DELIVERY_D
                              , I_QTY
                              ,
TOTAL_PRICE
                              ,
OL_DIST_INFO
                              , I_PRICE,
I_NAME, I_DATA, S_DATA,
S_QUANTITY

                        FROM DATA

                  ) AS INS
  ;
```

```
  // Start processing

  in_neword = (struct
in_neword_struct *)  pin ;
  neword    = (struct
out_neword_struct *) pout ;

#ifdef DEBUGIT
  new_debug( neword, in_neword,
"SP upon entry");
#endif

  // Using I_PRICE == 0 as a flag to
the client that the ITEM was not
fetched (hence bad).

  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", __FILE__, __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", __FILE__,
__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", __FILE__,
__LINE__, &sqlca);
          goto ferror;
      }
    }

    for ( inputItemArrayIndex = 0 ;
            inputItemArrayIndex <
in_neword->s_O_OL_CNT  // from
input
          &&
i_priceArray[ inputItemArrayIndex ] !=
0 ;
          inputItemArrayIndex++ )
    {
      // s_I_NAME, and s_S_QUANTITY
already set as output host variables

      neword-
>item[ inputItemArrayIndex ].s_I_PRIC
E =
i_priceArray[ inputItemArrayIndex ] ;

      if
(    is_ORIGINAL( s_dataArray[ inputIt
emArrayIndex ].data,
s_dataArray[ inputItemArrayIndex ].len
 )
            &&
is_ORIGINAL( i_dataArray[ inputItemA
rrayIndex ].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  )
      {
        neword->s_transtatus =
TRAN_OK ;

        EXEC SQL COMMIT;

        if( sqlca.sqlcode != 0 )
        {
          sqlerror(NEWORD_SQL,
"COMMIT", __FILE__, __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 FAILED (INVALID
ITEM)", __FILE__, __LINE__ , &sqlca);
          // no point in ferror
        }
      }
    }
    else
    {
      DLCHK( retry_tran );

      sqlerror( NEWORD_SQL,
"NEW_WH", __FILE__, __LINE__ ,
&sqlca);
      goto ferror;
    }

    /*-----------------------------------------------*/
    /*  Return to client                             */
    /*-----------------------------------------------*/

mexit:

    if ( sqlca.sqlcode >= 0 )
    {
      storedProcRc =
SQLZ_HOLD_PROC ;
    }
    else
    {
      storedProcRc =
SQLZ_DISCONNECT_PROC ;
    }

#ifdef DEBUGIT
    new_debug( neword, in_neword,
"SP prior to return");
#endif

    return ( storedProcRc ) ;

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 ,
__FILE__, __LINE__, &sqlca ) ;
  }

ferror:

  neword->s_transtatus =
FATAL_SQLERROR;

  EXEC SQL ROLLBACK WORK;

  if ( sqlca.sqlcode != 0 )
  {
    sqlerror( NEWORD_SQL,
"ROLLBACK FAILED", __FILE__,
__LINE__ , &sqlca ) ;
  }


  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 ) ;
}

// ---------------------------------------------
---------------
// Order Status SERVER
// ---------------------------------------------
---------------

#undef w_id
#undef d_id
#undef c_id_input
#undef o_id
#undef o_entry_d
#undef o_carrier_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 ] ;
    sqlint64 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     storedProcRc ;
  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 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 ;

#ifdef DEBUGIT
   ord_debug(ordstat, in_ordstat, "SP
upon entry");
#endif

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", __FILE__,
__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",
__FILE__, __LINE__ , &sqlca ) ;
      goto ferror;
   }

   itemArrayIndex = 0 ;
   {
      do
      {
         EXEC SQL FETCH
read_orderline_cur


INTO :ol_i_id , :ol_supply_w_id , :ol_q
uantity , :ol_amount , :ol_delivery_d ;

         if ( sqlca.sqlcode == 0 )
         {
            ordstat-
>item[ itemArrayIndex ].s_OL_I_ID
= ol_i_id ;
            ordstat-
>item[ itemArrayIndex ].s_OL_SUPPL
Y_W_ID     = ol_supply_w_id ;
            ordstat-
>item[ itemArrayIndex ].s_OL_QUANT
ITY        = ol_quantity ;
            ordstat-
>item[ itemArrayIndex ].s_OL_AMOU
NT          = ol_amount ;
            ordstat-
>item[ itemArrayIndex ].s_OL_DELIVE
RY_D_time = ol_delivery_d ;

            itemArrayIndex++;
         }
         else
         if (sqlca.sqlcode < 0 )
         {
            DLCHK( retry_tran ) ;
            sqlerror( ORDSTAT_SQL,
"FETCH CURSOR
read_orderline_cur" , __FILE__,
__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", __FILE__, __LINE__ ,
&sqlca);
      goto ferror ;
   }

mexit:

   if ( sqlca.sqlcode >= 0 )
   {
      storedProcRc =
SQLZ_HOLD_PROC ;
   }
   else
   {
      storedProcRc =
SQLZ_DISCONNECT_PROC ;
   }

#ifdef DEBUGIT
   ord_debug(ordstat, in_ordstat, "SP
prior to return");
#endif

   return ( storedProcRc ) ;

ferror:

   ordstat->s_transtatus =
FATAL_SQLERROR ;

   EXEC SQL ROLLBACK WORK ;

   if ( sqlca.sqlcode != 0 )
   {
      sqlerror(ORDSTAT_SQL,
"ROLLBACK FAILED", __FILE__,
__LINE__, &sqlca);
   }

   goto mexit;
}

// ---------------------------------------------
---------------
// Delivery SERVER
// ---------------------------------------------
---------------

#undef d_id
#undef c_id
#undef w_id
#undef o_carrier_id
#undef ol_delivery_d
```

```
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 storedProcRc ;

   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 ;

#ifdef DEBUGIT
   del_debug( delivery, in_delivery, "SP
upon entry");
#endif

   d_id = 1;

retry_tran:

   delivery->deadlocks++;

   for ( ;  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 )
      {
         delivery->s_O_ID[ d_id - 1 ] =
no_o_id ;
      }
      else
      {
         DLCHK( retry_tran );

         sqlerror( DELIVERY_SQL ,
"DELIVERY", __FILE__, __LINE__ ,
&sqlca);
         goto ferror ;
      }
   }

   delivery->s_transtatus = TRAN_OK ;

mexit:

   if ( sqlca.sqlcode >= 0 )
   {
      storedProcRc =
SQLZ_HOLD_PROC ;
   }
   else
   {
      storedProcRc =
SQLZ_DISCONNECT_PROC ;
   }

#ifdef DEBUGIT
   del_debug( delivery, in_delivery, "SP
prior to return");
#endif

   return ( storedProcRc ) ;

ferror:

   delivery->s_transtatus =
FATAL_SQLERROR ;

   EXEC SQL ROLLBACK WORK ;

   if ( sqlca.sqlcode != 0 )
   {
      sqlerror( DELIVERY_SQL,
"ROLLBACK FAILED", __FILE__,
__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 ) ;
}
```

_Src.Srv/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 - 2005
-- 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 $
```

Src.Srv/uncat-proc.ddl

```
DROP PROCEDURE news

(varchar(270),varchar(662));

DROP PROCEDURE news;



DROP PROCEDURE pays;

DROP PROCEDURE ords

(varchar(42),varchar(446));

DROP PROCEDURE ords;

DROP PROCEDURE dels

(varchar(22),varchar(50));

DROP PROCEDURE dels;

DROP PROCEDURE stks

(varchar(18),varchar(14));

DROP PROCEDURE stks;
```

include/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 - 2005
** 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.            */
/* Analysis indicates that a C_LAST
delta of 85 is optimal.                */
/*********************************************
********************************/
#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 CLIENT_SQL   0
#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

/*
*******************************************
************************** */
/* DB2 Connect/Disconnect & Thread
Context Wrappers                    */
/*
*******************************************
************************** */

#ifdef __cplusplus
extern "C" {
#endif

extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*,
char*, char*);
extern int disconnect_from_TM(void);
#ifdef __cplusplus
}
#endif

#endif // __DB2TPCC_H
```

_include/lval.h

```
#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 16800
#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
```

_include/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 - 2005
** 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"

#ifdef SWAP_ENDIAN
#define SWAP_BYTE(Var)
SwapEndian((void*)&Var, sizeof(Var))
```

```
/*****************************************
***********************************
  FUNCTION: SwapEndian
  PURPOSE: Swap the byte order of a
structure
  EXAMPLE: int I=0x12345678;
SWAP_BYTE(I); I => 0x78563412;
  IMPLEMENTATION: Fold Addr in
half, swap header & tail by XOR op
          e.g.: *a = 0x12 [ Addr + 0];
                *b = 0x78 [ Add + 4 - 0 - 1
= Addr+3];
                *a ^= *b;         // sets *a to
0x6A
                *b ^= *a;         // sets *b to
0x12
                *a ^= *b;         // sets *a to
0x78

           Now *a => 0x78 && *b =>
0x12
*****************************************
**********************************/

void SwapEndian(void *Addr, int nb)
{
  int i;
  for (i=0; i<nb/2; i++)
  {
    char *a = (char*)Addr+i;
    char *b = (char*)Addr+(nb-i-1);

    *a ^= *b;
    *b ^= *a;
    *a ^= *b;
  }
}
#endif   //SWAP_ENDIAN


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

include/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 - 2005
** 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

#ifdef __cplusplus
extern "C" {
#endif

extern void sqlerror (int tranType, char
*msg, char *file, int line,
                    SQL_STRUCTURE
sqlca *psqlca);

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);

#ifdef __cplusplus
}
#endif

#endif // __TPCCDBG_H
```

_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 - 2005
## 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 Kit Version
export TPCC_VERSION=CK041012

# 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}/func
tion
export TPCC_FENCED=NO
```

Appendix - B: Tunable Parameters

Database Parameters

db2set.cfg.out

```
DB2LINUXAIO=YES
DB2_SELUDI_COMM_BUFFER=YES
DB2_USE_ALTERNATE_PAGE_CLE
ANING=ON
DB2_MAX_NON_TABLE_LOCKS=50
0
DB2_LGPAGE_BP=ON
DB2_TRUSTED_BINDIN=ON
DB2_KEEPTABLELOCK=ON
DB2_NO_FORK_CHECK=ON
DB2_APM_PERFORMANCE=ALL
DB2_ENABLE_BUFPD=OFF
DB2_SELECTIVITY=YES
DB2ASSUMEUPDATE=ON
DB2CHECKCLIENTINTERVAL=0
DB2_HASH_JOIN=OFF
DB2_COLLECT_TS_REC_INFO=fals
e
DB2COMM=TCPIP
DB2CHKPTR=OFF
```

db.cfg.out

```
      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
 Database collating sequence
= BINARY
 Alternate collating sequence
(ALT_COLLATE) =
 Database page size
= 4096

 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
 Default maintained table types for opt
(DFT_MTTB_TYPES) = SYSTEM
 Number of frequent values retained
(NUM_FREQVALUES) = 10
 Number of quantiles retained
(NUM_QUANTILES) = 20

 Backup pending
= NO

 Database is consistent
= YES
 Rollforward pending
= NO
 Restore pending
= NO

 Multi-page file allocation enabled
= YES

 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) = 20000
 Size of database shared memory
(4KB)  (DATABASE_MEMORY) =
31600000
 Catalog cache size (4KB)
(CATALOGCACHE_SZ) =
(MAXAPPLS*4)
 Log buffer size (4KB)
(LOGBUFSZ) = 10000
 Utilities heap size (4KB)
(UTIL_HEAP_SZ) = 5000
 Buffer pool size (pages)
(BUFFPAGE) = 1000
 Extended storage segments size
(4KB)    (ESTORE_SEG_SZ) = 16000
 Number of extended storage
segments    (NUM_ESTORE_SEGS) =
0
 Max storage for lock list (4KB)
(LOCKLIST) = 40000

 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) = 4096
 Default application heap (4KB)
(APPLHEAPSZ) = 1500
 Package cache size (4KB)
(PCKCACHESZ) = 50000
 Statistics heap size (4KB)
(STAT_HEAP_SZ) = 4384

 Interval for checking deadlock (ms)
(DLCHKTIME) = 3000
 Percent. of lock lists per application
(MAXLOCKS) = 20
 Lock timeout (sec)
(LOCKTIMEOUT) = -1

 Changed pages threshold
(CHNGPGS_THRESH) = 99
 Number of asynchronous page
cleaners    (NUM_IOCLEANERS) = 10
 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) =
AUTOMATIC

 Track modified pages
(TRACKMOD) = OFF

 Default number of containers
= 1
 Default tablespace extentsize (pages)
(DFT_EXTENT_SZ) = 32

 Max number of active applications
(MAXAPPLS) = 1000
 Average number of active applications
(AVG_APPLS) = 1
 Max DB files open per application
(MAXFILOP) = 800

 Log file size (4KB)
(LOGFILSIZ) = 262144
 Number of primary log files
(LOGPRIMARY) = 125
 Number of secondary log files
(LOGSECOND) = 0
 Changed path to log files
(NEWLOGPATH) =
 Path to log files
= /dev/raw/raw251
 Overflow log path
(OVERFLOWLOGPATH) =
 Mirror log path
(MIRRORLOGPATH) =
 First active log file
= S0000112.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) = 1
```

```
 Percent log file reclaimed before soft
chckpt (SOFTMAX) = 1290
 Log retain for recovery enabled
(LOGRETAIN) = RECOVERY
 User exit for logging enabled
(USEREXIT) = OFF

 HADR database role
= STANDARD
 HADR local host name
(HADR_LOCAL_HOST) =
 HADR local service name
(HADR_LOCAL_SVC) =
 HADR remote host name
(HADR_REMOTE_HOST) =
 HADR remote service name
(HADR_REMOTE_SVC) =
 HADR instance name of remote
server  (HADR_REMOTE_INST) =
 HADR timeout value
(HADR_TIMEOUT) = 120
 HADR log write synchronization mode
(HADR_SYNCMODE) = NEARSYNC

 First log archive method
(LOGARCHMETH1) = LOGRETAIN
 Options for logarchmeth1
(LOGARCHOPT1) =
 Second log archive method
(LOGARCHMETH2) = OFF
 Options for logarchmeth2
(LOGARCHOPT2) =
 Failover log archive path
(FAILARCHPATH) =
 Number of log archive retries on error
(NUMARCHRETRY) = 5
 Log archive retry Delay (secs)
(ARCHRETRYDELAY) = 20
 Vendor options
(VENDOROPT) =

 Auto restart enabled
(AUTORESTART) = ON
 Index re-creation time and redo index
build  (INDEXREC) = SYSTEM
(RESTART)
 Log pages during index build
(LOGINDEXBUILD) = OFF
 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) =

 Automatic maintenance
(AUTO_MAINT) = OFF
   Automatic database backup
(AUTO_DB_BACKUP) = OFF
   Automatic table maintenance
(AUTO_TBL_MAINT) = OFF
     Automatic runstats
(AUTO_RUNSTATS) = OFF
     Automatic statistics profiling
(AUTO_STATS_PROF) = OFF
       Automatic profile updates
(AUTO_PROF_UPD) = OFF
     Automatic reorganization
(AUTO_REORG) = OFF
```

dbm.cfg.out

```
          Database Manager
Configuration

     Node type = Enterprise Server
Edition with local and remote clients

 Database manager configuration
release level            = 0x0a00

 CPU speed (millisec/instruction)
(CPUSPEED) = 3.778755e-07
 Communications bandwidth (MB/sec)
(COMM_BANDWIDTH) =
1.000000e+02

 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) = /opt/IBMJava2-ppc64-
142

 Diagnostic error capture level
(DIAGLEVEL) = 1
 Notify Level
(NOTIFYLEVEL) = 3
 Diagnostic data directory path
(DIAGPATH) =

 Default database monitor switches
   Buffer pool
(DFT_MON_BUFPOOL) = OFF
   Lock
(DFT_MON_LOCK) = OFF
   Sort
(DFT_MON_SORT) = OFF
   Statement
(DFT_MON_STMT) = OFF
   Table
(DFT_MON_TABLE) = OFF
   Timestamp
(DFT_MON_TIMESTAMP) = OFF
   Unit of work
(DFT_MON_UOW) = OFF
 Monitor health of instance and
databases   (HEALTH_MON) = OFF

 SYSADM group name
(SYSADM_GROUP) = DB2GRP1
 SYSCTRL group name
(SYSCTRL_GROUP) =
 SYSMAINT group name
(SYSMAINT_GROUP) =
 SYSMON group name
(SYSMON_GROUP) =

 Client Userid-Password Plugin
(CLNT_PW_PLUGIN) =
 Client Kerberos Plugin
(CLNT_KRB_PLUGIN) =
 Group Plugin
(GROUP_PLUGIN) =
 GSS Plugin for Local Authorization
(LOCAL_GSSPLUGIN) =
 Server Plugin Mode
(SRV_PLUGIN_MODE) =
UNFENCED
 Server List of GSS Plugins
(SRVCON_GSSPLUGIN_LIST) =
 Server Userid-Password Plugin
(SRVCON_PW_PLUGIN) =
 Server Connection Authentication
(SRVCON_AUTH) =
NOT_SPECIFIED
 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

 Workload impact by throttled
utilities(UTIL_IMPACT_LIM) = 10

 Priority of agents
(AGENTPRI) = 60
 Max number of existing agents
(MAXAGENTS) = 1000
```

Agent pool size
(NUM_POOLAGENTS) = 1
Initial number of agents in pool
(NUM_INITAGENTS) = 0
Max number of coordinating agents
(MAX_COORDAGENTS) =
(MAXAGENTS - NUM_INITAGENTS)
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
Initial number of fenced processes
(NUM_INITFENCED) = 0

Index re-creation time and redo index
build (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) = xtpcc
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
Number of FCM request blocks
(FCM_NUM_RQB) = AUTOMATIC
Number of FCM connection entries
(FCM_NUM_CONNECT) =
AUTOMATIC
Number of FCM message anchors
(FCM_NUM_ANCHORS) =
AUTOMATIC

Node connection elapse time (sec)
(CONN_ELAPSE) = 10
Max number of node connection
retries (MAX_CONNRETRIES) = 5
Max time difference between nodes
(min) (MAX_TIME_DIFF) = 60

db2start/db2stop timeout (min)
(START_STOP_TIME) = 10

Transaction Monitor Parameters

tpccCom.tpcc_com_settings.txt

Transactions not supported
Enable object pooling
Minimum pool size 24
Maximum pool size 24
Creation timeout 100,000
Enable Object Construction
Enable Just in time activation
Concurrency Required

InetInfo_registry.reg

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\InetInfo\Parameters]
"ListenBackLog"=dword:000000fa
"DispatchEntries"=hex(7):4c,00,44,00,41,00,
50,00,53,00,56,00,43,00,00,00,00,00
"MaxConnections"=dword:000061a8
"PoolThreadLimit"=dword:00000190
"ThreadTimeout"=dword:00015180
"MaxConcurrency"=dword:ffffffff

tcpip_parameters_registry.reg

Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters]
"NV Hostname"="client32"
"DataBasePath"=hex(2):25,00,53,00,79,00,
73,00,74,00,65,00,6d,00,52,00,6f,00,6f,\

00,74,00,25,00,5c,00,53,00,79,00,73,00,74,
00,65,00,6d,00,33,00,32,00,5c,00,\

64,00,72,00,69,00,76,00,65,00,72,00,73,00,
5c,00,65,00,74,00,63,00,00,00
"NameServer"=""
"ForwardBroadcasts"=dword:00000000
"IPEnableRouter"=dword:00000000
"Domain"=""
"Hostname"="client32"
"SearchList"=""
"UseDomainNameDevolution"=dword:00000
001
"EnableICMPRedirect"=dword:00000001
"DeadGWDetectDefault"=dword:00000001
"DontAddDefaultGatewayDefault"=dword:00
000000
"EnableSecurityFilters"=dword:00000000
"AllowUnqualifiedQuery"=dword:00000000
"PrioritizeRecordData"=dword:00000001
"GlobalTcpWindowSize"=dword:00008000
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Ad
apters]
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Ad
apters\NdisWanIp]
"LLInterface"="WANARP"
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,
70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,
49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,36,
00,31,00,44,00,33,00,33,00,39,00,\

37,00,37,00,2d,00,37,00,31,00,30,00,46,00,
2d,00,34,00,41,00,37,00,45,00,2d,\

00,38,00,44,00,34,00,37,00,2d,00,34,00,42,
00,34,00,38,00,32,00,42,00,43,00,\

35,00,32,00,33,00,46,00,38,00,7d,00,00,00,
54,00,63,00,70,00,69,00,70,00,5c,\

00,50,00,61,00,72,00,61,00,6d,00,65,00,74,
00,65,00,72,00,73,00,5c,00,49,00,\

6e,00,74,00,65,00,72,00,66,00,61,00,63,00,
65,00,73,00,5c,00,7b,00,39,00,41,\

00,33,00,41,00,41,00,36,00,41,00,43,00,2d,
00,35,00,38,00,34,00,36,00,2d,00,\

34,00,30,00,37,00,45,00,2d,00,38,00,32,00,
35,00,30,00,2d,00,46,00,30,00,33,\

00,42,00,36,00,30,00,34,00,39,00,36,00,36,
00,44,00,43,00,7d,00,00,00,00,00
"NumInterfaces"=dword:00000002
"IpInterfaces"=hex:77,39,d3,61,0f,71,7e,4a,
8d,47,4b,48,2b,c5,23,f8,ac,a6,3a,9a,\
46,58,7e,40,82,50,f0,3b,60,49,66,dc
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Ad
apters\{0435C97F-9186-473F-B181-
5449A2CF0042}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,
70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,
49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,30,
00,34,00,33,00,35,00,43,00,39,00,\

37,00,46,00,2d,00,39,00,31,00,38,00,36,00,
2d,00,34,00,37,00,33,00,46,00,2d,\

00,42,00,31,00,38,00,31,00,2d,00,35,00,34,
00,34,00,39,00,41,00,32,00,43,00,\

46,00,30,00,30,00,34,00,32,00,7d,00,00,00,
00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Ad
apters\{1E07A95A-92A0-4836-BF73-
7AE38F8ACA07}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,
70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,
49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,31,
00,45,00,30,00,37,00,41,00,39,00,\

35,00,41,00,2d,00,39,00,32,00,41,00,30,00,
2d,00,34,00,38,00,33,00,36,00,2d,\

00,42,00,46,00,37,00,33,00,2d,00,37,00,41,
00,45,00,33,00,38,00,46,00,38,00,\

41,00,43,00,41,00,30,00,37,00,7d,00,00,00,
00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Ad
apters\{2EA04AA5-93A6-437F-9153-
2F6834D3B795}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,
70,00,5c,00,50,00,61,00,72,00,61,00,\

```
6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,
49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,32,
00,45,00,41,00,30,00,34,00,41,00,\

41,00,35,00,2d,00,39,00,33,00,41,00,36,00,
2d,00,34,00,33,00,37,00,46,00,2d,\

00,39,00,31,00,35,00,33,00,2d,00,32,00,46,
00,36,00,38,00,33,00,34,00,44,00,\

33,00,42,00,37,00,39,00,35,00,7d,00,00,00,
00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Ad
apters\{37430121-7BE3-4B55-8AAB-
D8AD09B2029C}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,
70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,
49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,33,
00,37,00,34,00,33,00,30,00,31,00,\

32,00,31,00,2d,00,37,00,42,00,45,00,33,00,
2d,00,34,00,42,00,35,00,35,00,2d,\

00,38,00,41,00,41,00,42,00,2d,00,44,00,38,
00,41,00,44,00,30,00,39,00,42,00,\

32,00,30,00,32,00,39,00,43,00,7d,00,00,00,
00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Ad
apters\{6FE29D81-59D5-4401-A77E-
BE3BC929B6E0}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,
70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,
49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,36,
00,46,00,45,00,32,00,39,00,44,00,\

38,00,31,00,2d,00,35,00,39,00,44,00,35,00,
2d,00,34,00,34,00,30,00,31,00,2d,\

00,41,00,37,00,37,00,45,00,2d,00,42,00,45,
00,33,00,42,00,43,00,39,00,32,00,\

39,00,42,00,36,00,45,00,30,00,7d,00,00,00,
00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Ad
apters\{7B215199-A3F3-4836-89A6-
390C5E70E801}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,
70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,
49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,37,
00,42,00,32,00,31,00,35,00,31,00,\

39,00,39,00,2d,00,41,00,33,00,46,00,33,00,
2d,00,34,00,38,00,33,00,36,00,2d,\

00,38,00,39,00,41,00,36,00,2d,00,33,00,39,
00,30,00,43,00,35,00,45,00,37,00,\

30,00,45,00,38,00,30,00,31,00,7d,00,00,00,
00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Ad
apters\{A32BB4A3-C9B2-4ADB-A65D-
18BB314BF7F0}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,
70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,
49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,41,
00,33,00,32,00,42,00,42,00,34,00,\

41,00,33,00,2d,00,43,00,39,00,42,00,32,00,
2d,00,34,00,41,00,44,00,42,00,2d,\

00,41,00,36,00,35,00,44,00,2d,00,31,00,38,
00,42,00,42,00,33,00,31,00,34,00,\

42,00,46,00,37,00,46,00,30,00,7d,00,00,00,
00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Ad
apters\{A71EB7B5-37C6-42DB-BE8F-
BB231FD1BE00}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,
70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,
49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,41,
00,37,00,31,00,45,00,42,00,37,00,\

42,00,35,00,2d,00,33,00,37,00,43,00,36,00,
2d,00,34,00,32,00,44,00,42,00,2d,\

00,42,00,45,00,38,00,46,00,2d,00,42,00,42,
00,32,00,33,00,31,00,46,00,44,00,\

31,00,42,00,45,00,30,00,30,00,7d,00,00,00,
00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Ad
apters\{BEABCC14-9C0A-4BE9-9817-
14C4092418D3}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,
70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,
49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,42,
00,45,00,41,00,42,00,43,00,43,00,\

31,00,34,00,2d,00,39,00,43,00,30,00,41,00,
2d,00,34,00,42,00,45,00,39,00,2d,\

00,39,00,38,00,31,00,37,00,2d,00,31,00,34,
00,43,00,34,00,30,00,39,00,32,00,\

34,00,31,00,38,00,44,00,33,00,7d,00,00,00,
00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Ad
apters\{CD3F7746-9E60-4E22-9A40-
7BC6CC6B2E2E}]
"LLInterface"=""
"IpConfig"=hex(7):54,00,63,00,70,00,69,00,
70,00,5c,00,50,00,61,00,72,00,61,00,\

6d,00,65,00,74,00,65,00,72,00,73,00,5c,00,
49,00,6e,00,74,00,65,00,72,00,66,\

00,61,00,63,00,65,00,73,00,5c,00,7b,00,43,
00,44,00,33,00,46,00,37,00,37,00,\

34,00,36,00,2d,00,39,00,45,00,36,00,30,00,
2d,00,34,00,45,00,32,00,32,00,2d,\

00,39,00,41,00,34,00,30,00,2d,00,37,00,42,
00,43,00,36,00,43,00,43,00,36,00,\

42,00,32,00,45,00,32,00,45,00,7d,00,00,00,
00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\DN
SRegisteredAdapters]
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Int
erfaces]
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Int
erfaces\{0435C97F-9186-473F-B181-
5449A2CF0042}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,33,00,35,00,2e,0
0,31,00,2e,00,31,00,2e,00,31,00,00,00,\
  00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,
00,32,00,35,00,35,00,2e,00,32,00,35,\
  00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=
dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,
00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,
00
"RawIPAllowedProtocols"=hex(7):30,00,00,
00,00,00
"NTEContextList"=hex(7):00,00
"TcpWindowSize"=dword:00008000
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Int
erfaces\{1E07A95A-92A0-4836-BF73-
7AE38F8ACA07}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,0
0,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,
00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=
dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,
00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,
00
"RawIPAllowedProtocols"=hex(7):30,00,00,
00,00,00
"NTEContextList"=hex(7):00,00
"DhcpIPAddress"="0.0.0.0"
"DhcpSubnetMask"="255.0.0.0"
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Int
erfaces\{2EA04AA5-93A6-437F-9153-
2F6834D3B795}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,39,00,32,00,2e,0
0,31,00,36,00,38,00,2e,00,31,00,31,00,\
  2e,00,35,00,31,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,
00,32,00,35,00,35,00,2e,00,32,00,35,\
  00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
```

```
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=
dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,
00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,
00
"RawIPAllowedProtocols"=hex(7):30,00,00,
00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00
,30,00,30,00,30,00,30,00,30,00,30,00,\
  33,00,00,00,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:40b39c06
"T1"=dword:40b3a30e
"T2"=dword:40b3a854
"LeaseTerminatesTime"=dword:40b3aa16
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:0000000
0
"AddressType"=dword:00000000
"DhcpClassIdBin"=hex:
"TcpWindowSize"=dword:00008000
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Int
erfaces\{37430121-7BE3-4B55-8AAB-
D8AD09B2029C}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,0
0,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,
00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=
dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,
00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,
00
"RawIPAllowedProtocols"=hex(7):30,00,00,
00,00,00
"NTEContextList"=hex(7):00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Int
erfaces\{61D33977-710F-4A7E-8D47-
4B482BC523F8}]
"UseZeroBroadcast"=dword:00000000
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,0
0,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,
00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"EnableDeadGWDetect"=dword:00000001
"DontAddDefaultGateway"=dword:0000000
0
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Int
erfaces\{6FE29D81-59D5-4401-A77E-
BE3BC929B6E0}]
"UseZeroBroadcast"=dword:00000000

"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,0
0,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,
00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=
dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,
00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,
00
"RawIPAllowedProtocols"=hex(7):30,00,00,
00,00,00
"NTEContextList"=hex(7):00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Int
erfaces\{7B215199-A3F3-4836-89A6-
390C5E70E801}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,0
0,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,
00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=
dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,
00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,
00
"RawIPAllowedProtocols"=hex(7):30,00,00,
00,00,00
"NTEContextList"=hex(7):00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Int
erfaces\{9A3AA6AC-5846-407E-8250-
F03B604966DC}]
"UseZeroBroadcast"=dword:00000000
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,0
0,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,
00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"EnableDeadGWDetect"=dword:00000001
"DontAddDefaultGateway"=dword:0000000
0
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Int
erfaces\{A32BB4A3-C9B2-4ADB-A65D-
18BB314BF7F0}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,31,0
0,2e,00,33,00,2e,00,31,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,
00,32,00,35,00,35,00,2e,00,32,00,35,\
  00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=
dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,
00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,
00
"RawIPAllowedProtocols"=hex(7):30,00,00,
00,00,00
"NTEContextList"=hex(7):00,00
"TcpWindowSize"=dword:00008000
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Int
erfaces\{A71EB7B5-37C6-42DB-BE8F-
BB231FD1BE00}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,33,0
0,32,00,2e,00,31,00,2e,00,32,00,00,00,\
  00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,
00,32,00,35,00,35,00,2e,00,32,00,35,\
  00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=
dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,
00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,
00
"RawIPAllowedProtocols"=hex(7):30,00,00,
00,00,00
"NTEContextList"=hex(7):30,00,78,00,30,00
,30,00,30,00,30,00,30,00,30,00,30,00,\
  32,00,00,00,00,00
"DhcpServer"="255.255.255.255"
"Lease"=dword:00000e10
"LeaseObtainedTime"=dword:40be0640
"T1"=dword:40be0d48
"T2"=dword:40be128e
"LeaseTerminatesTime"=dword:40be1450
"IPAutoconfigurationAddress"="0.0.0.0"
"IPAutoconfigurationMask"="255.255.0.0"
"IPAutoconfigurationSeed"=dword:0000000
0
"AddressType"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Int
erfaces\{BEABCC14-9C0A-4BE9-9817-
14C4092418D3}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000000
"IPAddress"=hex(7):31,00,30,00,2e,00,31,0
0,2e,00,32,00,2e,00,31,00,00,00,00,00
"SubnetMask"=hex(7):32,00,35,00,35,00,2e,
00,32,00,35,00,35,00,2e,00,32,00,35,\
  00,35,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=
dword:00000000
"InterfaceMetric"=dword:00000001
"TCPAllowedPorts"=hex(7):30,00,00,00,00,
00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,
00
"RawIPAllowedProtocols"=hex(7):30,00,00,
00,00,00
"NTEContextList"=hex(7):00,00
"TcpWindowSize"=dword:00008000
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Int
erfaces\{CD3F7746-9E60-4E22-9A40-
7BC6CC6B2E2E}]
"UseZeroBroadcast"=dword:00000000
"EnableDeadGWDetect"=dword:00000001
"EnableDHCP"=dword:00000001
"IPAddress"=hex(7):30,00,2e,00,30,00,2e,0
0,30,00,2e,00,30,00,00,00,00,00
"SubnetMask"=hex(7):30,00,2e,00,30,00,2e,
00,30,00,2e,00,30,00,00,00,00,00
"DefaultGateway"=hex(7):00,00
"DefaultGatewayMetric"=hex(7):00,00
"NameServer"=""
"Domain"=""
"DisableDynamicUpdate"=dword:00000000
"EnableAdapterDomainNameRegistration"=
dword:00000000
"InterfaceMetric"=dword:00000001
```

```
"TCPAllowedPorts"=hex(7):30,00,00,00,00,
00
"UDPAllowedPorts"=hex(7):30,00,00,00,00,
00
"RawIPAllowedProtocols"=hex(7):30,00,00,
00,00,00
"NTEContextList"=hex(7):00,00
"TcpWindowSize"=dword:00008000
"DhcpIPAddress"="0.0.0.0"
"DhcpSubnetMask"="255.0.0.0"
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Per
sistentRoutes]
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\Tcpip\Parameters\Wi
nsock]
"UseDelayedAcceptance"=dword:00000000
"HelperDllName"=hex(2):25,00,53,00,79,00,
73,00,74,00,65,00,6d,00,52,00,6f,00,\

6f,00,74,00,25,00,5c,00,53,00,79,00,73,00,7
4,00,65,00,6d,00,33,00,32,00,5c,\

00,77,00,73,00,68,00,74,00,63,00,70,00,69,
00,70,00,2e,00,64,00,6c,00,6c,00,\
  00,00
"MaxSockAddrLength"=dword:00000010
"MinSockAddrLength"=dword:00000010
"Mapping"=hex:0b,00,00,00,03,00,00,00,02,
00,00,00,01,00,00,00,06,00,00,00,02,\

00,00,00,01,00,00,00,00,00,00,00,02,00,00,
00,00,00,00,00,06,00,00,00,00,00,\

00,00,00,00,00,00,06,00,00,00,00,00,00,00,
01,00,00,00,06,00,00,00,02,00,00,\

00,02,00,00,00,11,00,00,00,02,00,00,00,02,
00,00,00,00,00,00,00,02,00,00,00,\

00,00,00,00,11,00,00,00,00,00,00,00,00,00,
00,00,11,00,00,00,00,00,00,00,02,\

00,00,00,11,00,00,00,02,00,00,00,03,00,00,
00,00,00,00,00
```

Tpcc_software_registry.reg

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\TP
CC]
"dlvyLogPath"="c:\\InetPub\\wwwroot\\tpcc\\
dlvy"
"dlvyQueueLen"=dword:00004e20
"nullDB"=dword:00000000
"htmlTrace"=dword:00000000
"dbName"="tpcc"
"errorLogFile"="c:\\inetpub\\wwwroot\\tpcc\\i
sapi_err.log"
"htmlTraceLogFile"="c:\\inetpub\\wwwroot\\t
pcc\\isapi.log"
"numUsers"=dword:00005208
"dbType"="DB2"
"dbUserName"="tpcc"
"dbPassword"="tpcc"
"dbInterfacePath"="C:\\inetpub\\wwwroot\\tp
cc\\db2glue.dll"
"dlvyThreads"=dword:00000006
"isapi_trace"=dword:00000000
```

W3SVC_registry.reg

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000001
"ImagePath"=hex(2):43,00,3a,00,5c,00,57,0
0,49,00,4e,00,4e,00,54,00,5c,00,53,00,\

79,00,73,00,74,00,65,00,6d,00,33,00,32,00,
5c,00,69,00,6e,00,65,00,74,00,73,\

00,72,00,76,00,5c,00,69,00,6e,00,65,00,74,
00,69,00,6e,00,66,00,6f,00,2e,00,\
  65,00,78,00,65,00,00,00
"DisplayName"="World Wide Web
Publishing Service"
"DependOnService"=hex(7):49,00,49,00,53,
00,41,00,44,00,4d,00,49,00,4e,00,00,00,\
  00,00
"DependOnGroup"=hex(7):00,00
"ObjectName"="LocalSystem"
"Description"="Provides Web connectivity
and administration through the Internet
Information Services snap-in."
"FailureActions"=hex:ff,ff,ff,ff,00,00,00,00,00
,00,00,00,03,00,00,00,38,c3,0f,\

00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\W3SVC\ASP]
"NOTE"="This is for backward compatibility
only."
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\W3SVC\ASP\Parame
ters]
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\W3SVC\Parameters]
"MajorVersion"=dword:00000005
"MinorVersion"=dword:00000000
"InstallPath"="C:\\WINNT\\System32\\inetsrv
"
"CertMapList"="C:\\WINNT\\System32\\inets
rv\\iiscrmap.dll"
"AccessDeniedMessage"="Error: Access is
Denied."
"Filter DLLs"=""
"LogFileDirectory"="C:\\WINNT\\System32\\
LogFiles"
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\W3SVC\Parameters\
ADCLaunch]
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\W3SVC\Parameters\
ADCLaunch\AdvancedDataFactory]
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\W3SVC\Parameters\
ADCLaunch\RDSServer.DataFactory]
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\W3SVC\Parameters\
Script Map]
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\W3SVC\Parameters\
Virtual Roots]
"/"="c:\\inetpub\\wwwroot,,201"
"/Scripts"="c:\\inetpub\\scripts,,204"
"/IISHelp"="c:\\winnt\\help\\iishelp,,201"
"/IISAdmin"="C:\\WINNT\\System32\\inetsrv\
\iisadmin,,201"
"/IISSamples"="c:\\inetpub\\iissamples,,201"
"/MSADC"="c:\\program files\\common
files\\system\\msadc,,205"
"/_vti_bin"="C:\\Program Files\\Common
Files\\Microsoft Shared\\Web Server
Extensions\\40\\isapi,,205"
"/Printers"="C:\\WINNT\\web\\printers,,201"
"/tpcc"="C:\\Inetpub\\wwwroot\\tpcc,,207"
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\W3SVC\Performance
]
"Library"="w3ctrs.dll"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:000008e6
"Last Help"=dword:000008e7
"First Counter"=dword:00000844
"First Help"=dword:00000845
"Library Validation
Code"=hex:4a,82,e5,79,9f,41,c4,01,10,3d,0
0,00,00,00,00,00
"WbemAdapFileTime"=hex:00,9a,a9,c0,5f,3f
,c4,01
"WbemAdapFileSize"=dword:00003d10
"WbemAdapStatus"=dword:00000000
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\W3SVC\Security]
"Security"=hex:01,00,14,80,a0,00,00,00,ac,
00,00,00,14,00,00,00,30,00,00,00,02,\

00,1c,00,01,00,00,00,02,80,14,00,ff,01,0f,00
,01,01,00,00,00,00,00,01,00,00,\

00,00,02,00,70,00,04,00,00,00,00,00,18,00,
fd,01,02,00,01,01,00,00,00,00,00,\

05,12,00,00,00,74,00,6f,00,00,00,1c,00,ff,01
,0f,00,01,02,00,00,00,00,00,05,\

20,00,00,00,20,02,00,00,72,00,73,00,00,00,
18,00,8d,01,02,00,01,01,00,00,00,\

00,00,05,0b,00,00,00,20,02,00,00,00,00,1c,
00,fd,01,02,00,01,02,00,00,00,00,\

00,05,20,00,00,00,23,02,00,00,72,00,73,00,
01,01,00,00,00,00,00,05,12,00,00,\

  00,01,01,00,00,00,00,00,05,12,00,00,00
[HKEY_LOCAL_MACHINE\SYSTEM\Curre
ntControlSet\Services\W3SVC\Enum]
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

Linux Parameters

alloc_hugepages.sh

```
#!/bin/bash

echo 7830 >
/proc/sys/vm/nr_hugepages
```

chkconfig.txt

```
network         0:off   1:off   2:on   3:on
  4:on   5:on   6:off
iprinit         0:off   1:off   2:on   3:off
  4:on   5:on   6:off
vsftpd          0:off   1:off   2:off   3:on
  4:off   5:off   6:off
sendmail        0:off   1:off   2:on   3:off
  4:on   5:on   6:off
gpm             0:off   1:off   2:on   3:off
  4:on   5:on   6:off
kudzu           0:off   1:off   2:off   3:on
  4:on   5:on   6:off
rpcgssd         0:off   1:off   2:off   3:off
  4:on   5:on   6:off
rhnsd           0:off   1:off   2:off   3:off
  4:on   5:on   6:off
pcmcia          0:off   1:off   2:on   3:off
  4:on   5:on   6:off
atd             0:off   1:off   2:off   3:on
  4:on   5:on   6:off
dc_client       0:off   1:off   2:off   3:off
  4:off   5:off   6:off
```

```
cpuspeed      0:off  1:on  2:on  3:on
  4:on  5:on  6:off
bluetooth     0:off  1:off  2:off  3:off
  4:off  5:off  6:off
isdn          0:off  1:off  2:on  3:off
  4:on  5:on  6:off
rawdevices      0:off  1:off  2:off  3:on
  4:on  5:on  6:off
netplugd      0:off  1:off  2:off  3:off
  4:off  5:off  6:off
crond         0:off  1:off  2:on  3:on
  4:on  5:on  6:off
psacct        0:off  1:off  2:off  3:off
  4:off  5:off  6:off
saslauthd     0:off  1:off  2:off  3:off
  4:off  5:off  6:off
syslog        0:off  1:off  2:on  3:on
  4:on  5:on  6:off
xfs           0:off  1:off  2:on  3:off
  4:on  5:on  6:off
mdmonitor       0:off  1:off  2:on  3:off
  4:on  5:on  6:off
ypbind        0:off  1:off  2:off  3:off
  4:off  5:off  6:off
iptables      0:off  1:off  2:on  3:on
  4:on  5:on  6:off
autofs        0:off  1:off  2:off  3:on
  4:on  5:on  6:off
anacron       0:off  1:off  2:on  3:off
  4:on  5:on  6:off
diskdump      0:off  1:off  2:off  3:off
  4:off  5:off  6:off
cups          0:off  1:off  2:on  3:off
  4:on  5:on  6:off
ntpd          0:off  1:off  2:off  3:off
  4:off  5:off  6:off
nscd          0:off  1:off  2:off  3:off
  4:off  5:off  6:off
haldaemon       0:off  1:off  2:off  3:on
  4:on  5:on  6:off
rpcidmapd       0:off  1:off  2:off  3:off
  4:on  5:on  6:off
iprupdate     0:off  1:off  2:on  3:off
  4:on  5:on  6:off
xinetd        0:off  1:off  2:off  3:on
  4:on  5:on  6:off
irda          0:off  1:off  2:off  3:off
  4:off  5:off  6:off
NetworkManager  0:off  1:off  2:off
  3:off  4:off  5:off  6:off
readahead       0:off  1:off  2:off  3:off
  4:off  5:on  6:off
dc_server     0:off  1:off  2:off  3:off
  4:off  5:off  6:off
netdump       0:off  1:off  2:off  3:off
  4:off  5:off  6:off
sshd          0:off  1:off  2:on  3:on
  4:on  5:on  6:off
rpcsvcgssd      0:off  1:off  2:off  3:off
  4:on  5:on  6:off
mdmpd         0:off  1:off  2:off  3:off
  4:off  5:off  6:off
iprdump       0:off  1:off  2:on  3:off
  4:on  5:on  6:off
portmap       0:off  1:off  2:off  3:on
  4:on  5:on  6:off
smartd        0:off  1:off  2:on  3:on
  4:on  5:on  6:off
netfs         0:off  1:off  2:off  3:on
  4:on  5:on  6:off
messagebus    0:off  1:off  2:off  3:on
  4:on  5:on  6:off
readahead_early  0:off  1:off  2:off
  3:off  4:off  5:on  6:off
apmd          0:off  1:off  2:on  3:on
  4:on  5:on  6:off
nfs           0:off  1:off  2:off  3:on
  4:off  5:off  6:off
nfslock       0:off  1:off  2:off  3:on
  4:on  5:on  6:off
winbind       0:off  1:off  2:off  3:off
  4:off  5:off  6:off
xinetd based services:
  rsh: on
  daytime:  off
  krb5-telnet: off
  time:  off
  daytime-udp:  off
  gssftp:  off
  echo: off
  klogin:  off
  echo-udp: off
  time-udp: off
  chargen:  off
  rlogin:  off
  chargen-udp:  off
  eklogin: off
  rexec:  off
  rsync: off
  kshell:  off
  cups-lpd:  off
```

cmdline.txt

```
root=/dev/VolGroup00/LogVol00 ro
console=hvc0 elevator=noop
```

modprobe.conf

```
alias eth0 e1000
alias eth1 e1000
alias scsi_hostadapter ipr
alias scsi_hostadapter1 lpfc
alias usb-controller ehci-hcd
alias usb-controller1 ohci-hcd
options lpfc lpfc_cr_count=5
lpfc_cr_delay=1
options e1000
InterruptThrottleRate=1000,1000
```

setfifo.sh

```
#!/bin/ksh

./setsched -p 97 `ps -fu tpcc | grep
db2logg| awk '{print $2}'`
./setsched -p 96 `ps -fu tpcc | grep
db2ag | grep TPCC | awk '{print $2}'`
./setsched -p 96 `ps -fu tpcc | grep
db2pclnr | awk '{print $2}'`
```

setsched.c

```
#include <sched.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>

int main (int argc, char *argv[])
{
  pid_t pidList[2048];
  struct sched_param schedParam;
  int policy;
  int priority = 60;
  int i=1, pidIdx=0, j;

  while ( i < argc ) {
    if (strcmp(argv[i], "-p") == 0) {
      priority = atoi(argv[++i]);
    } else {
      pidList[pidIdx] = (pid_t)
atoi(argv[i]);
      pidIdx++;
    }
    i++;
  }

  for (j=0; j<pidIdx; j++) {
    sched_getparam(pidList[j],
&schedParam);

schedParam.sched_priority=priority;
    /* Set policy of SCHED_FIFO */
    sched_setscheduler(pidList[j],
SCHED_FIFO, &schedParam);

policy=sched_getscheduler(pidList[j]);
    printf("%d: policy = %d \t priority =
%d\n", (int)pidList[j], policy,
schedParam.sched_priority);
  }

  return(0);
}
```

sysctl.conf

```
# Kernel sysctl configuration file for
Red Hat Linux
#
# For binary values, 0 is disabled, 1 is
enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 0

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1

# Do not accept source routing
net.ipv4.conf.default.accept_source_ro
ute = 0

# Controls the System Request
debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will
append the PID to the core filename.
# Useful for debugging multi-threaded
applications.
kernel.core_uses_pid = 1

kernel.msgmni = 2048
kernel.shmmax = 1073741824
kernel.shmall = 137438953472
fs.file-max = 524288
net.core.rmem_max = 524288
net.core.wmem_max = 524288
kernel.sem = 1000        900000  128
4096
```

version.txt

Linux version 2.6.9-5.EL
(bhcompile@sizzler.build.redhat.com)
(gcc version 3.4.3 20041212 (Red Hat
3.4.3-9.EL4)) #1 SMP Wed Jan 5
19:23:58 EST 2005

Appendix - C: Database Setup Code

Database Creation Scripts

DDL/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_order_01
prefetchsize 0;
alter tablespace is_order_02
prefetchsize 0;
alter tablespace is_order_03
prefetchsize 0;
alter tablespace is_order_04
prefetchsize 0;
alter tablespace is_order_05
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_dis prefetchsize 0;
alter tablespace ts_history
prefetchsize 0;
alter tablespace ts_item_01
prefetchsize 0;
alter tablespace ts_newordaA_01
prefetchsize 0;
alter tablespace ts_newordaA_02
prefetchsize 0;
alter tablespace ts_newordaA_03
prefetchsize 0;
alter tablespace ts_newordaA_04
prefetchsize 0;
alter tablespace ts_newordaA_05
prefetchsize 0;
alter tablespace ts_newordaB_01
prefetchsize 0;
alter tablespace ts_newordaB_02
prefetchsize 0;
alter tablespace ts_newordaB_03
prefetchsize 0;
alter tablespace ts_newordaB_04
prefetchsize 0;
alter tablespace ts_newordaB_05
prefetchsize 0;
alter tablespace ts_order_01
prefetchsize 0;
alter tablespace ts_order_02
prefetchsize 0;
alter tablespace ts_order_03
prefetchsize 0;
alter tablespace ts_order_04
prefetchsize 0;
alter tablespace ts_order_05
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_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_wh prefetchsize 0;
connect reset;
```

DDL/ALTTBSP_PF_4096.ddl

```
connect to TPCC;
alter tablespace is_customer_01
prefetchsize 4096;
alter tablespace is_customer_02
prefetchsize 4096;
alter tablespace is_customer_03
prefetchsize 4096;
alter tablespace is_customer_04
prefetchsize 4096;
alter tablespace is_customer_05
prefetchsize 4096;
alter tablespace is_customer_06
prefetchsize 4096;
alter tablespace is_customer_07
prefetchsize 4096;
alter tablespace is_customer_08
prefetchsize 4096;
alter tablespace is_customer_09
prefetchsize 4096;
alter tablespace is_customer_10
prefetchsize 4096;
alter tablespace is_order_01
prefetchsize 4096;
alter tablespace is_order_02
prefetchsize 4096;
alter tablespace is_order_03
prefetchsize 4096;
alter tablespace is_order_04
prefetchsize 4096;
alter tablespace is_order_05
prefetchsize 4096;
alter tablespace ts_customer_01
prefetchsize 4096;
alter tablespace ts_customer_02
prefetchsize 4096;
alter tablespace ts_customer_03
prefetchsize 4096;
alter tablespace ts_customer_04
prefetchsize 4096;
alter tablespace ts_customer_05
prefetchsize 4096;
alter tablespace ts_customer_06
prefetchsize 4096;
alter tablespace ts_customer_07
prefetchsize 4096;
alter tablespace ts_customer_08
prefetchsize 4096;
alter tablespace ts_customer_09
prefetchsize 4096;
alter tablespace ts_customer_10
prefetchsize 4096;
alter tablespace ts_dis prefetchsize
4096;
alter tablespace ts_history
prefetchsize 4096;
alter tablespace ts_item_01
prefetchsize 4096;
alter tablespace ts_newordaA_01
prefetchsize 4096;
alter tablespace ts_newordaA_02
prefetchsize 4096;
alter tablespace ts_newordaA_03
prefetchsize 4096;
alter tablespace ts_newordaA_04
prefetchsize 4096;
alter tablespace ts_newordaA_05
prefetchsize 4096;
alter tablespace ts_newordaB_01
prefetchsize 4096;
alter tablespace ts_newordaB_02
prefetchsize 4096;
alter tablespace ts_newordaB_03
prefetchsize 4096;
alter tablespace ts_newordaB_04
prefetchsize 4096;
alter tablespace ts_newordaB_05
prefetchsize 4096;
alter tablespace ts_order_01
prefetchsize 4096;
```

```
alter tablespace ts_order_02
prefetchsize 4096;
alter tablespace ts_order_03
prefetchsize 4096;
alter tablespace ts_order_04
prefetchsize 4096;
alter tablespace ts_order_05
prefetchsize 4096;
alter tablespace ts_orderline_01
prefetchsize 4096;
alter tablespace ts_orderline_02
prefetchsize 4096;
alter tablespace ts_orderline_03
prefetchsize 4096;
alter tablespace ts_orderline_04
prefetchsize 4096;
alter tablespace ts_orderline_05
prefetchsize 4096;
alter tablespace ts_stock_01
prefetchsize 4096;
alter tablespace ts_stock_02
prefetchsize 4096;
alter tablespace ts_stock_03
prefetchsize 4096;
alter tablespace ts_stock_04
prefetchsize 4096;
alter tablespace ts_stock_05
prefetchsize 4096;
alter tablespace ts_stock_06
prefetchsize 4096;
alter tablespace ts_stock_07
prefetchsize 4096;
alter tablespace ts_stock_08
prefetchsize 4096;
alter tablespace ts_stock_09
prefetchsize 4096;
alter tablespace ts_stock_10
prefetchsize 4096;
alter tablespace ts_wh prefetchsize
4096;
connect reset;
```

DDL/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
1680);
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 1681
AND 3360);
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 3361
AND 5040);
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 5041
AND 6720);
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 6721
AND 8400);
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 8401
AND 10080);
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 10081
AND 11760);
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 11761
AND 13440);
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 13441
AND 15120);
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 >= 15121);
SET INTEGRITY FOR CUSTOMER10
ALL IMMEDIATE UNCHECKED;
connect reset;
```

DDL**/CRCONST_NEW_ORDERA.ddl**

```
connect to TPCC in share mode;
SET INTEGRITY FOR
NEW_ORDERA1 OFF;
ALTER TABLE NEW_ORDERA1
DROP CONSTRAINT
NEW_ORDERA1CKC;
ALTER TABLE NEW_ORDERA1 ADD
CONSTRAINT NEW_ORDERA1CKC
CHECK ((NO_W_ID BETWEEN 1
AND 3360) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR
NEW_ORDERA1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR
NEW_ORDERA2 OFF;
ALTER TABLE NEW_ORDERA2
DROP CONSTRAINT
NEW_ORDERA2CKC;
ALTER TABLE NEW_ORDERA2 ADD
CONSTRAINT NEW_ORDERA2CKC
CHECK ((NO_W_ID BETWEEN 3361
AND 6720) AND (NO_O_ID <= 3675));
SET INTEGRITY FOR
NEW_ORDERA2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR
NEW_ORDERA3 OFF;
ALTER TABLE NEW_ORDERA3
DROP CONSTRAINT
NEW_ORDERA3CKC;
ALTER TABLE NEW_ORDERA3 ADD
CONSTRAINT NEW_ORDERA3CKC
CHECK ((NO_W_ID BETWEEN 6721
AND 10080) AND (NO_O_ID <=
3675));
SET INTEGRITY FOR
NEW_ORDERA3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
```

```
SET INTEGRITY FOR
NEW_ORDERA4 OFF;
ALTER TABLE NEW_ORDERA4
DROP CONSTRAINT
NEW_ORDERA4CKC;
ALTER TABLE NEW_ORDERA4 ADD
CONSTRAINT NEW_ORDERA4CKC
CHECK ((NO_W_ID BETWEEN
10081 AND 13440) AND (NO_O_ID
<= 3675));
SET INTEGRITY FOR
NEW_ORDERA4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR
NEW_ORDERA5 OFF;
ALTER TABLE NEW_ORDERA5
DROP CONSTRAINT
NEW_ORDERA5CKC;
ALTER TABLE NEW_ORDERA5 ADD
CONSTRAINT NEW_ORDERA5CKC
CHECK ((NO_W_ID >= 13441) AND
(NO_O_ID <= 3675));
SET INTEGRITY FOR
NEW_ORDERA5 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

DDL/CRCONST_NEW_ORDERB.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR
NEW_ORDERB1 OFF;
ALTER TABLE NEW_ORDERB1
DROP CONSTRAINT
NEW_ORDERB1CKC;
ALTER TABLE NEW_ORDERB1 ADD
CONSTRAINT NEW_ORDERB1CKC
CHECK ((NO_W_ID BETWEEN 1
AND 3360) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR
NEW_ORDERB1 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR
NEW_ORDERB2 OFF;
ALTER TABLE NEW_ORDERB2
DROP CONSTRAINT
NEW_ORDERB2CKC;
ALTER TABLE NEW_ORDERB2 ADD
CONSTRAINT NEW_ORDERB2CKC
CHECK ((NO_W_ID BETWEEN 3361
AND 6720) AND (NO_O_ID >= 3676));
SET INTEGRITY FOR
NEW_ORDERB2 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR
NEW_ORDERB3 OFF;
ALTER TABLE NEW_ORDERB3
DROP CONSTRAINT
NEW_ORDERB3CKC;
ALTER TABLE NEW_ORDERB3 ADD
CONSTRAINT NEW_ORDERB3CKC
CHECK ((NO_W_ID BETWEEN 6721
AND 10080) AND (NO_O_ID >=
3676));
SET INTEGRITY FOR
NEW_ORDERB3 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR
NEW_ORDERB4 OFF;
ALTER TABLE NEW_ORDERB4
DROP CONSTRAINT
NEW_ORDERB4CKC;
ALTER TABLE NEW_ORDERB4 ADD
CONSTRAINT NEW_ORDERB4CKC
CHECK ((NO_W_ID BETWEEN
10081 AND 13440) AND (NO_O_ID
>= 3676));
SET INTEGRITY FOR
NEW_ORDERB4 ALL IMMEDIATE
UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR
NEW_ORDERB5 OFF;
ALTER TABLE NEW_ORDERB5
DROP CONSTRAINT
NEW_ORDERB5CKC;
ALTER TABLE NEW_ORDERB5 ADD
CONSTRAINT NEW_ORDERB5CKC
CHECK ((NO_W_ID >= 13441) AND
(NO_O_ID >= 3676));
SET INTEGRITY FOR
NEW_ORDERB5 ALL IMMEDIATE
UNCHECKED;
connect reset;
```

DDL/CRCONST_ORDERS.ddl

```
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS1
OFF;
ALTER TABLE ORDERS1 DROP
CONSTRAINT ORDERS1CKC;
ALTER TABLE ORDERS1 ADD
CONSTRAINT ORDERS1CKC
CHECK (O_W_ID BETWEEN 1 AND
3360);
SET INTEGRITY FOR ORDERS1 ALL
IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS2
OFF;
ALTER TABLE ORDERS2 DROP
CONSTRAINT ORDERS2CKC;
ALTER TABLE ORDERS2 ADD
CONSTRAINT ORDERS2CKC
CHECK (O_W_ID BETWEEN 3361
AND 6720);
SET INTEGRITY FOR ORDERS2 ALL
IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS3
OFF;
ALTER TABLE ORDERS3 DROP
CONSTRAINT ORDERS3CKC;
ALTER TABLE ORDERS3 ADD
CONSTRAINT ORDERS3CKC
CHECK (O_W_ID BETWEEN 6721
AND 10080);
SET INTEGRITY FOR ORDERS3 ALL
IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS4
OFF;
ALTER TABLE ORDERS4 DROP
CONSTRAINT ORDERS4CKC;
ALTER TABLE ORDERS4 ADD
CONSTRAINT ORDERS4CKC
CHECK (O_W_ID BETWEEN 10081
AND 13440);
SET INTEGRITY FOR ORDERS4 ALL
IMMEDIATE UNCHECKED;
connect reset;
connect to TPCC in share mode;
SET INTEGRITY FOR ORDERS5
OFF;
ALTER TABLE ORDERS5 DROP
CONSTRAINT ORDERS5CKC;
ALTER TABLE ORDERS5 ADD
CONSTRAINT ORDERS5CKC
CHECK (O_W_ID >= 13441);
SET INTEGRITY FOR ORDERS5 ALL
IMMEDIATE UNCHECKED;
connect reset;
```

DDL/CRCONST_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
3360);
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 3361
AND 6720);
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 6721
AND 10080);
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 10081
AND 13440);
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 >= 13441);
SET INTEGRITY FOR ORDER_LINE5
ALL IMMEDIATE UNCHECKED;
connect reset;
```

DDL/CRCONST_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 1680);
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 1681 AND 3360);
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 3361 AND 5040);
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 5041 AND 6720);
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 6721 AND 8400);
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 8401 AND
10080);
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 10081 AND
11760);
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 11761 AND
13440);
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 13441 AND
15120);
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 >= 15121);
SET INTEGRITY FOR STOCK10 ALL
IMMEDIATE UNCHECKED;
connect reset;
```

DDL/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;
```

DDL/CRIDX_ORDR_IDXB.ddl

```
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB1;
CREATE INDEX ORDR_IDXB1
                ON
ORDERS1(O_C_ID, O_W_ID,
O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
```

```
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB2;
CREATE INDEX ORDR_IDXB2
                    ON
ORDERS2(O_C_ID, O_W_ID,
O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB3;
CREATE INDEX ORDR_IDXB3
                    ON
ORDERS3(O_C_ID, O_W_ID,
O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB4;
CREATE INDEX ORDR_IDXB4
                    ON
ORDERS4(O_C_ID, O_W_ID,
O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
connect to TPCC in share mode;
DROP INDEX ORDR_IDXB5;
CREATE INDEX ORDR_IDXB5
                    ON
ORDERS5(O_C_ID, O_W_ID,
O_D_ID, O_ID DESC) PCTFREE 20
LEVEL2 PCTFREE 20;
connect reset;
```

DDL/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 1680,
               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 1681 ENDING AT 3360,
               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 3361 ENDING AT 5040,
               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 5041 ENDING AT 6720,
                 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 6721 ENDING AT 8400,
                 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 8401 ENDING AT 10080,
                 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 10081 ENDING AT 11760,
                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 11761 ENDING AT 13440,
                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 13441 ENDING AT 15120,
                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 15121 ENDING AT 16800,
                C_D_ID STARTING
FROM 1 ENDING AT 10
               )
               ALLOW OVERFLOW;
connect reset;
```

DDL/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_dis
               INDEX IN ts_dis
               ORGANIZE BY KEY
SEQUENCE (
                D_ID STARTING FROM 1
ENDING AT 10,
                D_W_ID STARTING
FROM 1 ENDING AT 16800
               )
               ALLOW OVERFLOW;
connect reset;
```

DDL/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
               INDEX IN ts_history;
ALTER TABLE HISTORY APPEND
ON;
connect reset;
```

DDL/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;
```

DDL/CRTB_NEW_ORDERA.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA1;
CREATE TABLE NEW_ORDERA1
               (
                NO_O_ID         INTEGER
NOT NULL,
                NO_D_ID
SMALLINT    NOT NULL,
                NO_W_ID         INTEGER
NOT NULL
               )
               IN ts_newordaA_01
               INDEX IN
ts_newordaA_01
               ORGANIZE BY KEY
SEQUENCE (
                NO_W_ID STARTING
FROM 1 ENDING AT 3360,
                NO_D_ID STARTING
FROM 1 ENDING AT 10,
                NO_O_ID STARTING
FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA2;
CREATE TABLE NEW_ORDERA2
               (
                NO_O_ID         INTEGER
NOT NULL,
                NO_D_ID
SMALLINT    NOT NULL,
                NO_W_ID         INTEGER
NOT NULL
               )
               IN ts_newordaA_02
               INDEX IN
ts_newordaA_02
               ORGANIZE BY KEY
SEQUENCE (
                NO_W_ID STARTING
FROM 3361 ENDING AT 6720,
                NO_D_ID STARTING
FROM 1 ENDING AT 10,
                NO_O_ID STARTING
FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA3;
CREATE TABLE NEW_ORDERA3
               (
                NO_O_ID         INTEGER
NOT NULL,
                NO_D_ID
SMALLINT    NOT NULL,
                NO_W_ID         INTEGER
NOT NULL
               )
               IN ts_newordaA_03
               INDEX IN
ts_newordaA_03
               ORGANIZE BY KEY
SEQUENCE (
                NO_W_ID STARTING
FROM 6721 ENDING AT 10080,
                NO_D_ID STARTING
FROM 1 ENDING AT 10,
                NO_O_ID STARTING
FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA4;
CREATE TABLE NEW_ORDERA4
               (
                NO_O_ID         INTEGER
NOT NULL,
                NO_D_ID
SMALLINT    NOT NULL,
                NO_W_ID         INTEGER
NOT NULL
               )
               IN ts_newordaA_04
               INDEX IN
ts_newordaA_04
               ORGANIZE BY KEY
SEQUENCE (
                NO_W_ID STARTING
FROM 10081 ENDING AT 13440,
                NO_D_ID STARTING
FROM 1 ENDING AT 10,
                NO_O_ID STARTING
FROM 1900 ENDING AT 3675
               )
               ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERA5;
CREATE TABLE NEW_ORDERA5
               (
                NO_O_ID         INTEGER
NOT NULL,
                NO_D_ID
SMALLINT    NOT NULL,
                NO_W_ID         INTEGER
NOT NULL
               )
               IN ts_newordaA_05
               INDEX IN
ts_newordaA_05
```

```
            ORGANIZE BY KEY
SEQUENCE (
             NO_W_ID STARTING
FROM 13441 ENDING AT 16800,
             NO_D_ID STARTING
FROM 1 ENDING AT 10,
             NO_O_ID STARTING
FROM 1900 ENDING AT 3675
            )
            ALLOW OVERFLOW;
connect reset;
```

DDL/CRTB_NEW_ORDERB.ddl

```
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB1;
CREATE TABLE NEW_ORDERB1
            (
             NO_O_ID        INTEGER
NOT NULL,
             NO_D_ID
SMALLINT    NOT NULL,
             NO_W_ID        INTEGER
NOT NULL
            )
            IN ts_newordaB_01
            INDEX IN
ts_newordaB_01
            ORGANIZE BY KEY
SEQUENCE (
             NO_W_ID STARTING
FROM 1 ENDING AT 3360,
             NO_D_ID STARTING
FROM 1 ENDING AT 10,
             NO_O_ID STARTING
FROM 3676 ENDING AT 5451
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB2;
CREATE TABLE NEW_ORDERB2
            (
             NO_O_ID        INTEGER
NOT NULL,
             NO_D_ID
SMALLINT    NOT NULL,
             NO_W_ID        INTEGER
NOT NULL
            )
            IN ts_newordaB_02
            INDEX IN
ts_newordaB_02
            ORGANIZE BY KEY
SEQUENCE (
             NO_W_ID STARTING
FROM 3361 ENDING AT 6720,
             NO_D_ID STARTING
FROM 1 ENDING AT 10,
             NO_O_ID STARTING
FROM 3676 ENDING AT 5451
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB3;
CREATE TABLE NEW_ORDERB3
            (
             NO_O_ID        INTEGER
NOT NULL,
             NO_D_ID
SMALLINT    NOT NULL,
             NO_W_ID        INTEGER
NOT NULL
            )
            IN ts_newordaB_03
            INDEX IN
ts_newordaB_03
            ORGANIZE BY KEY
SEQUENCE (
             NO_W_ID STARTING
FROM 6721 ENDING AT 10080,
             NO_D_ID STARTING
FROM 1 ENDING AT 10,
             NO_O_ID STARTING
FROM 3676 ENDING AT 5451
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB4;
CREATE TABLE NEW_ORDERB4
            (
             NO_O_ID        INTEGER
NOT NULL,
             NO_D_ID
SMALLINT    NOT NULL,
             NO_W_ID        INTEGER
NOT NULL
            )
            IN ts_newordaB_04
            INDEX IN
ts_newordaB_04
            ORGANIZE BY KEY
SEQUENCE (
             NO_W_ID STARTING
FROM 10081 ENDING AT 13440,
             NO_D_ID STARTING
FROM 1 ENDING AT 10,
             NO_O_ID STARTING
FROM 3676 ENDING AT 5451
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE NEW_ORDERB5;
CREATE TABLE NEW_ORDERB5
            (
             NO_O_ID        INTEGER
NOT NULL,
             NO_D_ID
SMALLINT    NOT NULL,
             NO_W_ID        INTEGER
NOT NULL
            )
            IN ts_newordaB_05
            INDEX IN
ts_newordaB_05
            ORGANIZE BY KEY
SEQUENCE (
             NO_W_ID STARTING
FROM 13441 ENDING AT 16800,
             NO_D_ID STARTING
FROM 1 ENDING AT 10,
             NO_O_ID STARTING
FROM 3676 ENDING AT 5451
            )
            ALLOW OVERFLOW;
connect reset;
```

DDL/CRTB_ORDERS.ddl

```
connect to TPCC in share mode;
DROP TABLE ORDERS1;
CREATE TABLE ORDERS1
            (
             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
0 ENDING AT 3675,
             O_W_ID STARTING
FROM 1 ENDING AT 3360,
             O_D_ID STARTING
FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS2;
CREATE TABLE ORDERS2
            (
             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_02
            INDEX IN is_order_02
            ORGANIZE BY KEY
SEQUENCE (
             O_ID STARTING FROM
0 ENDING AT 3675,
             O_W_ID STARTING
FROM 3361 ENDING AT 6720,
             O_D_ID STARTING
FROM 1 ENDING AT 10
            )
            ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS3;
```

```
CREATE TABLE ORDERS3
(
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_03
INDEX IN is_order_03
ORGANIZE BY KEY
SEQUENCE (
O_ID STARTING FROM
0 ENDING AT 3675,
O_W_ID STARTING
FROM 6721 ENDING AT 10080,
O_D_ID STARTING
FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS4;
CREATE TABLE ORDERS4
(
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_04
INDEX IN is_order_04
ORGANIZE BY KEY
SEQUENCE (
O_ID STARTING FROM
0 ENDING AT 3675,
O_W_ID STARTING
FROM 10081 ENDING AT 13440,
O_D_ID STARTING
FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
connect to TPCC in share mode;
DROP TABLE ORDERS5;
CREATE TABLE ORDERS5
(
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_05
INDEX IN is_order_05
ORGANIZE BY KEY
SEQUENCE (
O_ID STARTING FROM
0 ENDING AT 3675,
O_W_ID STARTING
FROM 13441 ENDING AT 16800,
O_D_ID STARTING
FROM 1 ENDING AT 10
)
ALLOW OVERFLOW;
connect reset;
```

DDL/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 3360,
OL_D_ID STARTING
FROM 1 ENDING AT 10,
OL_O_ID STARTING
FROM 0 ENDING AT 3675,
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 3361 ENDING AT 6720,
OL_D_ID STARTING
FROM 1 ENDING AT 10,
OL_O_ID STARTING
FROM 0 ENDING AT 3675,
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 6721 ENDING AT 10080,
```

```
                OL_D_ID STARTING
FROM 1 ENDING AT 10,
                OL_O_ID STARTING
FROM 0 ENDING AT 3675,
                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 10081 ENDING AT 13440,
                OL_D_ID STARTING
FROM 1 ENDING AT 10,
                OL_O_ID STARTING
FROM 0 ENDING AT 3675,
                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 13441 ENDING AT 16800,
                OL_D_ID STARTING
FROM 1 ENDING AT 10,
                OL_O_ID STARTING
FROM 0 ENDING AT 3675,
                OL_NUMBER STARTING
FROM 1 ENDING AT 15
               )
               ALLOW OVERFLOW;
connect reset;
```

DDL/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 1680
               )
               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 1681 ENDING AT 3360
               )
               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 3361 ENDING AT 5040
               )
               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 5041 ENDING AT 6720
               )
               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 6721 ENDING AT 8400
               )
               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 8401 ENDING AT 10080
               )
               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 10081 ENDING AT 11760
               )
               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 11761 ENDING AT 13440
               )
               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 13441 ENDING AT 15120
               )
               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 15121 ENDING AT 16800
               )
               ALLOW OVERFLOW;
connect reset;
```

DDL/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_wh
               INDEX IN ts_wh
               ORGANIZE BY KEY
SEQUENCE (
                W_ID STARTING FROM
1 ENDING AT 16800
               )
               ALLOW OVERFLOW;
connect reset;
```

DDL/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 UNION ALL
SELECT * FROM CUSTOMER2
UNION ALL
SELECT * FROM CUSTOMER3
UNION ALL
SELECT * FROM CUSTOMER4
UNION ALL
SELECT * FROM CUSTOMER5
UNION ALL
SELECT * FROM CUSTOMER6
UNION ALL
SELECT * FROM CUSTOMER7
UNION ALL
SELECT * FROM CUSTOMER8
UNION ALL
SELECT * FROM CUSTOMER9
UNION ALL
SELECT * FROM CUSTOMER10
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/CRVW_NEW_ORDER.ddl

```
connect to TPCC in share mode;
DROP VIEW NEW_ORDER;
CREATE VIEW NEW_ORDER
           (NO_O_ID,
            NO_D_ID,
            NO_W_ID
           ) AS SELECT * FROM
NEW_ORDERA1 UNION ALL
SELECT * FROM NEW_ORDERA2
UNION ALL
SELECT * FROM NEW_ORDERA3
UNION ALL
SELECT * FROM NEW_ORDERA4
UNION ALL
SELECT * FROM NEW_ORDERA5
UNION ALL
SELECT * FROM NEW_ORDERB1
UNION ALL
SELECT * FROM NEW_ORDERB2
UNION ALL
SELECT * FROM NEW_ORDERB3
UNION ALL
SELECT * FROM NEW_ORDERB4
UNION ALL
SELECT * FROM NEW_ORDERB5
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/CRVW_ORDERS.ddl

```
connect to TPCC in share mode;
DROP VIEW ORDERS;
CREATE VIEW ORDERS
           (O_C_ID,
            O_ENTRY_D,
            O_CARRIER_ID,
            O_OL_CNT,
            O_ALL_LOCAL,
            O_ID,
            O_W_ID,
            O_D_ID
           ) AS SELECT * FROM
ORDERS1 UNION ALL
SELECT * FROM ORDERS2 UNION
ALL
SELECT * FROM ORDERS3 UNION
ALL
SELECT * FROM ORDERS4 UNION
ALL
SELECT * FROM ORDERS5
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/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 UNION ALL
SELECT * FROM ORDER_LINE2
UNION ALL
SELECT * FROM ORDER_LINE3
UNION ALL
SELECT * FROM ORDER_LINE4
UNION ALL
SELECT * FROM ORDER_LINE5
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/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 UNION ALL
SELECT * FROM STOCK2 UNION
ALL
SELECT * FROM STOCK3 UNION
ALL
SELECT * FROM STOCK4 UNION
ALL
SELECT * FROM STOCK5 UNION
ALL
SELECT * FROM STOCK6 UNION
ALL
SELECT * FROM STOCK7 UNION
ALL
SELECT * FROM STOCK8 UNION
ALL
SELECT * FROM STOCK9 UNION
ALL
SELECT * FROM STOCK10
WITH ROW MOVEMENT;
COMMIT WORK;
connect reset;
```

DDL/GEN_CUSTOMER_ALL.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
7 -r 1 1680 -f1
/sdk14/f1_01/customer_01_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
7 -r 1681 3360 -f1
/sdk14/f1_02/customer_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
7 -r 3361 5040 -f1
/sdk14/f1_03/customer_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
7 -r 5041 6720 -f1
/sdk14/f1_04/customer_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
7 -r 6721 8400 -f1
/sdk14/f1_05/customer_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
7 -r 8401 10080 -f1
/sdk14/f1_06/customer_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
7 -r 10081 11760 -f1
/sdk14/f1_07/customer_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
7 -r 11761 13440 -f1
/sdk14/f1_08/customer_08_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
7 -r 13441 15120 -f1
/sdk14/f1_09/customer_09_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
7 -r 15121 16800 -f1
/sdk14/f1_10/customer_10_1.dat
```

DDL/GEN_DISTRICT_ALL.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
4 -r 1 16800 -f1 /sdf14/f1/district_1.dat
```

DDL/GEN_HISTORY_ALL.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
8 -r 1 16800 -f1 /sdf14/f1/history_1.dat
```

DDL/GEN_ITEM_ALL.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
5 -f1 /sdf14/f1/item_1.dat
```

DDL/GEN_NEW_ORDER_ALL.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
11 -r 1 3360 -f1
/sdf14/f1_01/newordera_01_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
11 -r 3361 6720 -f1
/sdf14/f1_02/newordera_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
11 -r 6721 10080 -f1
/sdf14/f1_03/newordera_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
11 -r 10081 13440 -f1
/sdf14/f1_04/newordera_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
11 -r 13441 16800 -f1
/sdf14/f1_05/newordera_05_1.dat
```

DDL/GEN_ORDERS_ALL.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
9 -r 1 3360 -f1
/sdf14/f1_01/orders_01_1.dat -f2
/sdf14/f1_01/orderline_01_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
9 -r 3361 6720 -f1
/sdf14/f1_02/orders_02_1.dat -f2
/sdf14/f1_02/orderline_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
9 -r 6721 10080 -f1
/sdf14/f1_03/orders_03_1.dat -f2
/sdf14/f1_03/orderline_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
9 -r 10081 13440 -f1
/sdf14/f1_04/orders_04_1.dat -f2
/sdf14/f1_04/orderline_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
9 -r 13441 16800 -f1
/sdf14/f1_05/orders_05_1.dat -f2
/sdf14/f1_05/orderline_05_1.dat
```

DDL/GEN_STOCK_ALL.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
6 -r 1 1680 -f1
/sdpu/f1_01/stock_01_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
6 -r 1681 3360 -f1
/sdpu/f1_02/stock_02_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
6 -r 3361 5040 -f1
/sdpu/f1_03/stock_03_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
6 -r 5041 6720 -f1
/sdpu/f1_04/stock_04_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
6 -r 6721 8400 -f1
/sdpu/f1_05/stock_05_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
6 -r 8401 10080 -f1
/sdpu/f1_06/stock_06_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
6 -r 10081 11760 -f1
/sdpu/f1_07/stock_07_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
6 -r 11761 13440 -f1
/sdpu/f1_08/stock_08_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
6 -r 13441 15120 -f1
/sdpu/f1_09/stock_09_1.dat
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
6 -r 15121 16800 -f1
/sdpu/f1_10/stock_10_1.dat
```

DDL/GEN_WAREHOUSE_ALL.sh

```
/home/tpcc/tpc-c.ibm/dbgen/gendata -t
3 -r 1 16800 -f1
/sdf14/f1/warehouse_1.dat
```

DDL/LOAD_CUSTOMER_ALL.ddl

```
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE CUSTOMER1
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdk14/f1_01/customer_01_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 50400000 INSERT
INTO CUSTOMER1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE CUSTOMER2
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdk14/f1_02/customer_02_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 50400000 INSERT
INTO CUSTOMER2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE CUSTOMER3
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdk14/f1_03/customer_03_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 50400000 INSERT
INTO CUSTOMER3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE CUSTOMER4
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdk14/f1_04/customer_04_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 50400000 INSERT
INTO CUSTOMER4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE CUSTOMER5
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdk14/f1_05/customer_05_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 50400000 INSERT
INTO CUSTOMER5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE CUSTOMER6
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdk14/f1_06/customer_06_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 50400000 INSERT
INTO CUSTOMER6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE CUSTOMER7
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdk14/f1_07/customer_07_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 50400000 INSERT
INTO CUSTOMER7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE CUSTOMER8
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdk14/f1_08/customer_08_1.dat OF
```

```
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 50400000 INSERT
INTO CUSTOMER8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE CUSTOMER9
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdk14/f1_09/customer_09_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 50400000 INSERT
INTO CUSTOMER9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE CUSTOMER10
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdk14/f1_10/customer_10_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 50400000 INSERT
INTO CUSTOMER10;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_DISTRICT_ALL.ddl

```
CONNECT TO TPCC IN SHARE
MODE;
IMPORT FROM /sdf14/f1/district_1.dat
OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO
DISTRICT;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_HISTORY_ALL.ddl

```
connect to TPCC in share mode;
LOAD FROM /sdf14/f1/history_1.dat
OF DEL MODIFIED BY COLDEL|
KEEPBLANKS FASTPARSE
REPLACE INTO HISTORY
NONRECOVERABLE DATA BUFFER
16000 CPU_PARALLELISM 4 ;
connect reset;
```

DDL/LOAD_ITEM_ALL.ddl

```
CONNECT TO TPCC IN SHARE
MODE;
IMPORT FROM /sdf14/f1/item_1.dat
OF DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 1000 INSERT INTO
ITEM;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_NEW_ORDER_ALL.ddl

```
CONNECT TO TPCC IN SHARE
MODE;
IMPORT FROM
/sdf14/f1_01/newordera_01_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS
COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
IMPORT FROM
/sdf14/f1_02/newordera_02_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS
COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
IMPORT FROM
/sdf14/f1_03/newordera_03_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS
COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
IMPORT FROM
/sdf14/f1_04/newordera_04_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS
COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
IMPORT FROM
/sdf14/f1_05/newordera_05_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS
COMMITCOUNT 20000 INSERT
INTO NEW_ORDERA5;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_ORDERS_ALL.ddl

```
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE ORDERS1 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdf14/f1_01/orders_01_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 100800000 INSERT
INTO ORDERS1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE ORDERS2 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdf14/f1_02/orders_02_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 100800000 INSERT
INTO ORDERS2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE ORDERS3 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdf14/f1_03/orders_03_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 100800000 INSERT
INTO ORDERS3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE ORDERS4 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdf14/f1_04/orders_04_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 100800000 INSERT
INTO ORDERS4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE ORDERS5 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdf14/f1_05/orders_05_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 100800000 INSERT
INTO ORDERS5;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_ORDER_LINE_ALL.ddl

```
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE ORDER_LINE1
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdf14/f1_01/orderline_01_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 1108800000
INSERT INTO ORDER_LINE1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE ORDER_LINE2
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdf14/f1_02/orderline_02_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 1108800000
INSERT INTO ORDER_LINE2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE ORDER_LINE3
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdf14/f1_03/orderline_03_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 1108800000
INSERT INTO ORDER_LINE3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE ORDER_LINE4
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdf14/f1_04/orderline_04_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 1108800000
INSERT INTO ORDER_LINE4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE ORDER_LINE5
ACTIVATE NOT LOGGED INITIALLY;
IMPORT FROM
/sdf14/f1_05/orderline_05_1.dat OF
DEL MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 1108800000
INSERT INTO ORDER_LINE5;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_STOCK_ALL.ddl

```
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE STOCK1 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdpu/f1_01/stock_01_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 168000000 INSERT
INTO STOCK1;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE STOCK2 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdpu/f1_02/stock_02_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 168000000 INSERT
INTO STOCK2;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE STOCK3 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdpu/f1_03/stock_03_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 168000000 INSERT
INTO STOCK3;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE STOCK4 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdpu/f1_04/stock_04_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 168000000 INSERT
INTO STOCK4;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE STOCK5 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdpu/f1_05/stock_05_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 168000000 INSERT
INTO STOCK5;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE STOCK6 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdpu/f1_06/stock_06_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 168000000 INSERT
INTO STOCK6;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE STOCK7 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdpu/f1_07/stock_07_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 168000000 INSERT
INTO STOCK7;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE STOCK8 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdpu/f1_08/stock_08_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 168000000 INSERT
INTO STOCK8;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE STOCK9 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdpu/f1_09/stock_09_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 168000000 INSERT
INTO STOCK9;
COMMIT WORK;
CONNECT RESET;
CONNECT TO TPCC IN SHARE
MODE;
UPDATE COMMAND OPTIONS
USING C OFF;
ALTER TABLE STOCK10 ACTIVATE
NOT LOGGED INITIALLY;
IMPORT FROM
/sdpu/f1_10/stock_10_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
COMMITCOUNT 168000000 INSERT
INTO STOCK10;
COMMIT WORK;
CONNECT RESET;
```

DDL/LOAD_WAREHOUSE_ALL.ddl

```
CONNECT TO TPCC IN SHARE
MODE;
IMPORT FROM
/sdf14/f1/warehouse_1.dat OF DEL
MODIFIED BY COLDEL|
KEEPBLANKS COMPOUND=50
ALLOW WRITE ACCESS
COMMITCOUNT 1000 INSERT INTO
WAREHOUSE;
COMMIT WORK;
CONNECT RESET;
```

DDL/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;
```

DDL/RNST_DISTRICT.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.DISTRICT AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_HISTORY.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.HISTORY AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_ITEM.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE tpcc.ITEM
AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_NEW_ORDERA.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.NEW_ORDERA1 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.NEW_ORDERA2 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.NEW_ORDERA3 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.NEW_ORDERA4 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.NEW_ORDERA5 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_NEW_ORDERB.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.NEW_ORDERB1 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.NEW_ORDERB2 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.NEW_ORDERB3 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.NEW_ORDERB4 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.NEW_ORDERB5 AND INDEXES
ALL;
COMMIT WORK;
connect reset;
```

DDL/RNST_ORDERS.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.ORDERS1 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.ORDERS2 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.ORDERS3 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.ORDERS4 AND INDEXES ALL;
COMMIT WORK;
connect reset;
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.ORDERS5 AND INDEXES ALL;
COMMIT WORK;
connect reset;
```

DDL/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;
```

DDL/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;
```

DDL/RNST_WAREHOUSE.ddl

```
connect to TPCC in share mode;
RUNSTATS ON TABLE
tpcc.WAREHOUSE AND INDEXES
ALL;
COMMIT WORK;
connect reset;
```

bp/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 IBMDEFAULTBP
DEFERRED SIZE      500;
alter bufferpool IBMDEFAULT8K
DEFERRED SIZE      500;
alter bufferpool IBMDEFAULT16K
DEFERRED SIZE      500;

alter bufferpool WH
DEFERRED SIZE      433;
alter bufferpool DIS          DEFERRED
SIZE     4600;
alter bufferpool ITEM
DEFERRED SIZE     1235;
alter bufferpool HST
DEFERRED SIZE      753;
alter bufferpool NEW
DEFERRED SIZE   480000;
alter bufferpool ORD
DEFERRED SIZE   180047;
alter bufferpool ORD_I
DEFERRED SIZE  1043687;
alter bufferpool OLN
DEFERRED SIZE   840000;
alter bufferpool CST
DEFERRED SIZE   440000;
alter bufferpool CST_I
DEFERRED SIZE   530000;
alter bufferpool STK
DEFERRED SIZE 24600000;

connect reset;
db2stop;
terminate;
```

bp/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 - 2005
-- 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 NOT EXTENDED
STORAGE;
CREATE BUFFERPOOL
"IBMDEFAULT16K"  SIZE 1000
PAGESIZE 16384 NOT EXTENDED
STORAGE;

CREATE BUFFERPOOL "WH"
SIZE 1000    PAGESIZE 4096 NOT
EXTENDED STORAGE;
CREATE BUFFERPOOL "DIS"    SIZE
1000    PAGESIZE 4096 NOT
EXTENDED STORAGE;
CREATE BUFFERPOOL "ITEM"
SIZE 1000    PAGESIZE 8192 NOT
EXTENDED STORAGE;
CREATE BUFFERPOOL "HST"
SIZE 1000    PAGESIZE 16384 NOT
EXTENDED STORAGE;
CREATE BUFFERPOOL "NEW"
SIZE 1000    PAGESIZE 4096 NOT
EXTENDED STORAGE;
CREATE BUFFERPOOL "ORD"
SIZE 1000    PAGESIZE 8192 NOT
EXTENDED STORAGE;
```

```
CREATE BUFFERPOOL "ORD_I"
SIZE 1000    PAGESIZE 8192 NOT
EXTENDED STORAGE;
CREATE BUFFERPOOL "OLN"
SIZE 1000    PAGESIZE 8192 NOT
EXTENDED STORAGE;
CREATE BUFFERPOOL "CST"
SIZE 1000    PAGESIZE 4096 NOT
EXTENDED STORAGE;
CREATE BUFFERPOOL "CST_I"
SIZE 1000    PAGESIZE 8192 NOT
EXTENDED STORAGE;
CREATE BUFFERPOOL "STK"
SIZE 100000  PAGESIZE 4096 NOT
EXTENDED STORAGE;


connect reset;
terminate;
```

db/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 - 2005
-- 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 /db_home
collate using identity
catalog tablespace
managed by system using
('/db_home/db1catalog');
```

ts/create_tablespace.ddl

```
connect to tpcc;

drop tablespace ts_wh;
create regular tablespace ts_wh
pagesize 4K
   managed by database
   using
  (
     device '/dev/raw/raw1'  70,
     device '/dev/raw/raw2'  70,
     device '/dev/raw/raw3'  70,
     device '/dev/raw/raw4'  70,
     device '/dev/raw/raw5'  70,
     device '/dev/raw/raw6'  70,
     device '/dev/raw/raw7'  70,
     device '/dev/raw/raw8'  70,
     device '/dev/raw/raw9'  70,
     device '/dev/raw/raw10' 70,
     device '/dev/raw/raw11' 70,
     device '/dev/raw/raw12' 70,
     device '/dev/raw/raw13' 70,
     device '/dev/raw/raw14' 70,
     device '/dev/raw/raw15' 70,
     device '/dev/raw/raw16' 70,
     device '/dev/raw/raw17' 70,
     device '/dev/raw/raw18' 70,
     device '/dev/raw/raw19' 70,
     device '/dev/raw/raw20' 70
          )
     extentsize 32
     bufferpool wh
     prefetchsize 4096;
commit;

drop tablespace ts_dis;
create regular tablespace ts_dis
pagesize 4K
     managed by database
     using
    (
          device '/dev/raw/raw21'
560,
          device '/dev/raw/raw22'
560,
          device '/dev/raw/raw23'
560,
          device '/dev/raw/raw24'
560,
          device '/dev/raw/raw25'
560,
          device '/dev/raw/raw26'
560,
          device '/dev/raw/raw27'
560,
          device '/dev/raw/raw28'
560,
          device '/dev/raw/raw29'
560,
          device '/dev/raw/raw30'
560,
          device '/dev/raw/raw31'
560,
          device '/dev/raw/raw32'
560,
          device '/dev/raw/raw33'
560,
          device '/dev/raw/raw34'
560,
          device '/dev/raw/raw35'
560,
          device '/dev/raw/raw36'
560,
          device '/dev/raw/raw37'
560,
          device '/dev/raw/raw38'
560,
          device '/dev/raw/raw39'
560,
          device '/dev/raw/raw40'
560
         )
     extentsize 64
     bufferpool dis
     prefetchsize 4096;
commit;

drop tablespace ts_item_01;
create regular tablespace ts_item_01
pagesize 8K
     managed by database
     using
    (
          device '/dev/raw/raw41'
180,
          device '/dev/raw/raw42'
180,
          device '/dev/raw/raw43'
180,
          device '/dev/raw/raw44'
180,
          device '/dev/raw/raw45'
180,
          device '/dev/raw/raw46'
180,
          device '/dev/raw/raw47'
180,
          device '/dev/raw/raw48'
180,
          device '/dev/raw/raw49'
180,
          device '/dev/raw/raw50'
180,
          device '/dev/raw/raw51'
180,
          device '/dev/raw/raw52'
180,
          device '/dev/raw/raw53'
180,
          device '/dev/raw/raw54'
180,
          device '/dev/raw/raw55'
180,
          device '/dev/raw/raw56'
180,
          device '/dev/raw/raw57'
180,
          device '/dev/raw/raw58'
180,
          device '/dev/raw/raw59'
180,
          device '/dev/raw/raw60'
180
         )
     extentsize 16
     bufferpool item
     prefetchsize 4096;
commit;

drop tablespace ts_stock_01;
create regular tablespace ts_stock_01
pagesize 4K
     managed by database
     using
    (
          device '/dev/raw/raw61'
7400000,
          device '/dev/raw/raw62'
7400000
         )
     extentsize 256
     bufferpool stk
     prefetchsize 4096;
commit;
drop tablespace ts_stock_02;
create regular tablespace ts_stock_02
pagesize 4K
     managed by database
     using
    (
```

```
            device '/dev/raw/raw63'
7400000,
            device '/dev/raw/raw64'
7400000
            )
      extentsize 256
      bufferpool stk
      prefetchsize 4096;
commit;
drop tablespace ts_stock_03;
create regular tablespace ts_stock_03
pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw65'
7400000,
            device '/dev/raw/raw66'
7400000
            )
      extentsize 256
      bufferpool stk
      prefetchsize 4096;
commit;
drop tablespace ts_stock_04;
create regular tablespace ts_stock_04
pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw67'
7400000,
            device '/dev/raw/raw68'
7400000
            )
      extentsize 256
      bufferpool stk
      prefetchsize 4096;
commit;
drop tablespace ts_stock_05;
create regular tablespace ts_stock_05
pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw69'
7400000,
            device '/dev/raw/raw70'
7400000
            )
      extentsize 256
      bufferpool stk
      prefetchsize 4096;
commit;
drop tablespace ts_stock_06;
create regular tablespace ts_stock_06
pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw71'
7400000,
            device '/dev/raw/raw72'
7400000
            )
      extentsize 256
      bufferpool stk
      prefetchsize 4096;
commit;
drop tablespace ts_stock_07;
create regular tablespace ts_stock_07
pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw73'
7400000,
            device '/dev/raw/raw74'
7400000
            )
      extentsize 256
      bufferpool stk
      prefetchsize 4096;
commit;
drop tablespace ts_stock_08;
create regular tablespace ts_stock_08
pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw75'
7400000,
            device '/dev/raw/raw76'
7400000
            )
      extentsize 256
      bufferpool stk
      prefetchsize 4096;
commit;
drop tablespace ts_stock_09;
create regular tablespace ts_stock_09
pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw77'
7400000,
            device '/dev/raw/raw78'
7400000
            )
      extentsize 256
      bufferpool stk
      prefetchsize 4096;
commit;
drop tablespace ts_stock_10;
create regular tablespace ts_stock_10
pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw79'
7400000,
            device '/dev/raw/raw80'
7400000
            )
      extentsize 256
      bufferpool stk
      prefetchsize 4096;
commit;

drop tablespace ts_customer_01;
create regular tablespace
ts_customer_01 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw81'
5300000,
            device '/dev/raw/raw82'
5300000
            )
      extentsize 256
      bufferpool cst
      prefetchsize 4096;
commit;
drop tablespace ts_customer_02;
create regular tablespace
ts_customer_02 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw83'
5300000,
            device '/dev/raw/raw84'
5300000
            )
      extentsize 256
      bufferpool cst
      prefetchsize 4096;
commit;
drop tablespace ts_customer_03;
create regular tablespace
ts_customer_03 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw85'
5300000,
            device '/dev/raw/raw86'
5300000
            )
      extentsize 256
      bufferpool cst
      prefetchsize 4096;
commit;
drop tablespace ts_customer_04;
create regular tablespace
ts_customer_04 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw87'
5300000,
            device '/dev/raw/raw88'
5300000
            )
      extentsize 256
      bufferpool cst
      prefetchsize 4096;
commit;
drop tablespace ts_customer_05;
create regular tablespace
ts_customer_05 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw89'
5300000,
            device '/dev/raw/raw90'
5300000
            )
      extentsize 256
      bufferpool cst
      prefetchsize 4096;
commit;
drop tablespace ts_customer_06;
create regular tablespace
ts_customer_06 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw91'
5300000,
            device '/dev/raw/raw92'
5300000
            )
```

```
        extentsize 256
        bufferpool cst
        prefetchsize 4096;
commit;
drop tablespace ts_customer_07;
create regular tablespace
ts_customer_07 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw93'
5300000,
                device '/dev/raw/raw94'
5300000
               )
        extentsize 256
        bufferpool cst
        prefetchsize 4096;
commit;
drop tablespace ts_customer_08;
create regular tablespace
ts_customer_08 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw95'
5300000,
                device '/dev/raw/raw96'
5300000
               )
        extentsize 256
        bufferpool cst
        prefetchsize 4096;
commit;
drop tablespace ts_customer_09;
create regular tablespace
ts_customer_09 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw97'
5300000,
                device '/dev/raw/raw98'
5300000
               )
        extentsize 256
        bufferpool cst
        prefetchsize 4096;
commit;
drop tablespace ts_customer_10;
create regular tablespace
ts_customer_10 pagesize 4K
         managed by database
         using
        (
                device '/dev/raw/raw99'
5300000,
                device '/dev/raw/raw100'
5300000
               )
        extentsize 256
        bufferpool cst
        prefetchsize 4096;
commit;

drop tablespace is_customer_01;
create regular tablespace
is_customer_01 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw101'
170000,
                device '/dev/raw/raw102'
170000
               )
        extentsize 256
        bufferpool cst_i
        prefetchsize 4096;
commit;
drop tablespace is_customer_02;
create regular tablespace
is_customer_02 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw103'
170000,
                device '/dev/raw/raw104'
170000
               )
        extentsize 256
        bufferpool cst_i
        prefetchsize 4096;
commit;
drop tablespace is_customer_03;
create regular tablespace
is_customer_03 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw105'
170000,
                device '/dev/raw/raw106'
170000
               )
        extentsize 256
        bufferpool cst_i
        prefetchsize 4096;
commit;
drop tablespace is_customer_04;
create regular tablespace
is_customer_04 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw107'
170000,
                device '/dev/raw/raw108'
170000
               )
        extentsize 256
        bufferpool cst_i
        prefetchsize 4096;
commit;
drop tablespace is_customer_05;
create regular tablespace
is_customer_05 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw109'
170000,
                device '/dev/raw/raw110'
170000
               )
        extentsize 256
        bufferpool cst_i
        prefetchsize 4096;
commit;
drop tablespace is_customer_06;
create regular tablespace
is_customer_06 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw111'
170000,
                device '/dev/raw/raw112'
170000
               )
        extentsize 256
        bufferpool cst_i
        prefetchsize 4096;
commit;
drop tablespace is_customer_07;
create regular tablespace
is_customer_07 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw113'
170000,
                device '/dev/raw/raw114'
170000
               )
        extentsize 256
        bufferpool cst_i
        prefetchsize 4096;
commit;
drop tablespace is_customer_08;
create regular tablespace
is_customer_08 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw115'
170000,
                device '/dev/raw/raw116'
170000
               )
        extentsize 256
        bufferpool cst_i
        prefetchsize 4096;
commit;
drop tablespace is_customer_09;
create regular tablespace
is_customer_09 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw117'
170000,
                device '/dev/raw/raw118'
170000
               )
        extentsize 256
        bufferpool cst_i
        prefetchsize 4096;
commit;
drop tablespace is_customer_10;
create regular tablespace
is_customer_10 pagesize 8K
         managed by database
         using
        (
                device '/dev/raw/raw119'
170000,
                device '/dev/raw/raw120'
170000
               )
        extentsize 256
        bufferpool cst_i
        prefetchsize 4096;
commit;
```

```
drop tablespace ts_history;
create regular tablespace ts_history
pagesize 16K
        managed by database
        using
       (
              device '/dev/raw/raw121'
150000,
              device '/dev/raw/raw122'
150000,
              device '/dev/raw/raw123'
150000,
              device '/dev/raw/raw124'
150000,
              device '/dev/raw/raw125'
150000,
              device '/dev/raw/raw126'
150000,
              device '/dev/raw/raw127'
150000,
              device '/dev/raw/raw128'
150000,
              device '/dev/raw/raw129'
150000,
              device '/dev/raw/raw130'
150000,
              device '/dev/raw/raw131'
150000,
              device '/dev/raw/raw132'
150000,
              device '/dev/raw/raw133'
150000,
              device '/dev/raw/raw134'
150000,
              device '/dev/raw/raw135'
150000,
              device '/dev/raw/raw136'
150000,
              device '/dev/raw/raw137'
150000,
              device '/dev/raw/raw138'
150000,
              device '/dev/raw/raw139'
150000,
              device '/dev/raw/raw140'
150000
             )
       extentsize 256
       bufferpool hst
       prefetchsize 4096;
commit;

drop tablespace ts_order_01;
create regular tablespace ts_order_01
pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw141'
155000,
              device '/dev/raw/raw142'
155000,
              device '/dev/raw/raw143'
155000,
              device '/dev/raw/raw144'
155000
             )
       extentsize 256
       bufferpool ord
       prefetchsize 4096;
commit;
drop tablespace ts_order_02;
create regular tablespace ts_order_02
pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw145'
155000,
              device '/dev/raw/raw146'
155000,
              device '/dev/raw/raw147'
155000,
              device '/dev/raw/raw148'
155000
             )
       extentsize 256
       bufferpool ord
       prefetchsize 4096;
commit;
drop tablespace ts_order_03;
create regular tablespace ts_order_03
pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw149'
155000,
              device '/dev/raw/raw150'
155000,
              device '/dev/raw/raw151'
155000,
              device '/dev/raw/raw152'
155000
             )
       extentsize 256
       bufferpool ord
       prefetchsize 4096;
commit;
drop tablespace ts_order_04;
create regular tablespace ts_order_04
pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw153'
155000,
              device '/dev/raw/raw154'
155000,
              device '/dev/raw/raw155'
155000,
              device '/dev/raw/raw156'
155000
             )
       extentsize 256
       bufferpool ord
       prefetchsize 4096;
commit;
drop tablespace ts_order_05;
create regular tablespace ts_order_05
pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw157'
155000,
              device '/dev/raw/raw158'
155000,
              device '/dev/raw/raw159'
155000,
              device '/dev/raw/raw160'
155000
             )
       extentsize 256
       bufferpool ord
       prefetchsize 4096;
commit;

drop tablespace is_order_01;
create regular tablespace is_order_01
pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw161'
140000,
              device '/dev/raw/raw162'
140000,
              device '/dev/raw/raw163'
140000,
              device '/dev/raw/raw164'
140000
             )
       extentsize 256
       bufferpool ord_i
       prefetchsize 4096;
commit;
drop tablespace is_order_02;
create regular tablespace is_order_02
pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw165'
140000,
              device '/dev/raw/raw166'
140000,
              device '/dev/raw/raw167'
140000,
              device '/dev/raw/raw168'
140000
             )
       extentsize 256
       bufferpool ord_i
       prefetchsize 4096;
commit;
drop tablespace is_order_03;
create regular tablespace is_order_03
pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw169'
140000,
              device '/dev/raw/raw170'
140000,
              device '/dev/raw/raw171'
140000,
              device '/dev/raw/raw172'
140000
             )
       extentsize 256
       bufferpool ord_i
       prefetchsize 4096;
commit;
drop tablespace is_order_04;
create regular tablespace is_order_04
pagesize 8K
        managed by database
        using
       (
              device '/dev/raw/raw173'
140000,
              device '/dev/raw/raw174'
140000,
```

```
            device '/dev/raw/raw175'
140000,
            device '/dev/raw/raw176'
140000
          )
      extentsize 256
      bufferpool ord_i
      prefetchsize 4096;
commit;
drop tablespace is_order_05;
create regular tablespace is_order_05
pagesize 8K
       managed by database
       using
      (
            device '/dev/raw/raw177'
140000,
            device '/dev/raw/raw178'
140000,
            device '/dev/raw/raw179'
140000,
            device '/dev/raw/raw180'
140000
          )
      extentsize 256
      bufferpool ord_i
      prefetchsize 4096;
commit;

drop tablespace ts_orderline_01;
create regular tablespace
ts_orderline_01 pagesize 8K
       managed by database
       using
      (
            device '/dev/raw/raw181'
4100000,
            device '/dev/raw/raw182'
4100000,
            device '/dev/raw/raw183'
4100000,
            device '/dev/raw/raw184'
4100000
          )
      extentsize 256
      bufferpool oln
      prefetchsize 4096;
commit;
drop tablespace ts_orderline_02;
create regular tablespace
ts_orderline_02 pagesize 8K
       managed by database
       using
      (
            device '/dev/raw/raw185'
4100000,
            device '/dev/raw/raw186'
4100000,
            device '/dev/raw/raw187'
4100000,
            device '/dev/raw/raw188'
4100000
          )
      extentsize 256
      bufferpool oln
      prefetchsize 4096;
commit;
drop tablespace ts_orderline_03;
create regular tablespace
ts_orderline_03 pagesize 8K
       managed by database
       using
      (
            device '/dev/raw/raw189'
4100000,
            device '/dev/raw/raw190'
4100000,
            device '/dev/raw/raw191'
4100000,
            device '/dev/raw/raw192'
4100000
          )
      extentsize 256
      bufferpool oln
      prefetchsize 4096;
commit;
drop tablespace ts_orderline_04;
create regular tablespace
ts_orderline_04 pagesize 8K
       managed by database
       using
      (
            device '/dev/raw/raw193'
4100000,
            device '/dev/raw/raw194'
4100000,
            device '/dev/raw/raw195'
4100000,
            device '/dev/raw/raw196'
4100000
          )
      extentsize 256
      bufferpool oln
      prefetchsize 4096;
commit;

drop tablespace ts_orderline_05;
create regular tablespace
ts_orderline_05 pagesize 8K
       managed by database
       using
      (
            device '/dev/raw/raw197'
4100000,
            device '/dev/raw/raw198'
4100000,
            device '/dev/raw/raw199'
4100000,
            device '/dev/raw/raw200'
4100000
          )
      extentsize 256
      bufferpool oln
      prefetchsize 4096;
commit;

drop tablespace ts_neworda_01;
create regular tablespace
ts_neworda_01 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw201'
120000,
            device '/dev/raw/raw202'
120000,
            device '/dev/raw/raw203'
120000,
            device '/dev/raw/raw204'
120000
          )
      extentsize 256
      bufferpool new
      prefetchsize 4096;
commit;

drop tablespace ts_neworda_02;
create regular tablespace
ts_neworda_02 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw205'
120000,
            device '/dev/raw/raw206'
120000,
            device '/dev/raw/raw207'
120000,
            device '/dev/raw/raw208'
120000
          )
      extentsize 256
      bufferpool new
      prefetchsize 4096;
commit;

drop tablespace ts_neworda_03;
create regular tablespace
ts_neworda_03 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw209'
120000,
            device '/dev/raw/raw210'
120000,
            device '/dev/raw/raw211'
120000,
            device '/dev/raw/raw212'
120000
          )
      extentsize 256
      bufferpool new
      prefetchsize 4096;
commit;

drop tablespace ts_neworda_04;
create regular tablespace
ts_neworda_04 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw213'
120000,
            device '/dev/raw/raw214'
120000,
            device '/dev/raw/raw215'
120000,
            device '/dev/raw/raw216'
120000
          )
      extentsize 256
      bufferpool new
      prefetchsize 4096;
commit;

drop tablespace ts_neworda_05;
create regular tablespace
ts_neworda_05 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw217'
120000,
            device '/dev/raw/raw218'
120000,
```

```
            device '/dev/raw/raw219'
120000,
            device '/dev/raw/raw220'
120000
          )
      extentsize 256
      bufferpool new
      prefetchsize 4096;
commit;

drop tablespace ts_newordb_01;
create regular tablespace
ts_newordb_01 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw221'
120000,
            device '/dev/raw/raw222'
120000,
            device '/dev/raw/raw223'
120000,
            device '/dev/raw/raw224'
120000
          )
      extentsize 256
      bufferpool new
      prefetchsize 4096;
commit;

drop tablespace ts_newordb_02;
create regular tablespace
ts_newordb_02 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw225'
120000,
            device '/dev/raw/raw226'
120000,
            device '/dev/raw/raw227'
120000,
            device '/dev/raw/raw228'
120000
          )
      extentsize 256
      bufferpool new
      prefetchsize 4096;
commit;

drop tablespace ts_newordb_03;
create regular tablespace
ts_newordb_03 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw229'
120000,
            device '/dev/raw/raw230'
120000,
            device '/dev/raw/raw231'
120000,
            device '/dev/raw/raw232'
120000
          )
      extentsize 256
      bufferpool new
      prefetchsize 4096;
commit;

drop tablespace ts_newordb_04;
create regular tablespace
ts_newordb_04 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw233'
120000,
            device '/dev/raw/raw234'
120000,
            device '/dev/raw/raw235'
120000,
            device '/dev/raw/raw236'
120000
          )
      extentsize 256
      bufferpool new
      prefetchsize 4096;
commit;

drop tablespace ts_newordb_05;
create regular tablespace
ts_newordb_05 pagesize 4K
       managed by database
       using
      (
            device '/dev/raw/raw237'
120000,
            device '/dev/raw/raw238'
120000,
            device '/dev/raw/raw239'
120000,
            device '/dev/raw/raw240'
120000
          )
      extentsize 256
      bufferpool new
      prefetchsize 4096;
commit;

connect reset;
```

SQL Creation

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 - 2005
## All Rights Reserved.
##
## US Government Users Restricted
Rights - Use, duplication or
## disclosure restricted by GSA ADP
Schedule Contract with IBM Corp.
################################
################################
###############

#
# Makefile.config - Linux 32/64-bit
#

# Make Configuration
MAKE=make

# Compiler Configuration.
# CFLAGS_DEBUG may be set to "-g",
"-DDEBUGIT" "-g -DDEBUGIT" or left
blank
CC=cc
CFLAGS_OS=-m64 -DSQLUNIX -
DSQLLinux -O3 -fpic
#CFLAGS_OS=-DDEBUGIT -
DSQLUNIX -DSQLLinux -O3 -fpic
CFLAGS_OUT=-o
CFLAGS_DEBUG=

# Linker Configuration
LD_EXEC=cc
LD_STORP=ld
LDFLAGS_EXEC=-m64
LDFLAGS_STORP=-shared -m
elf64ppc
LDFLAGS_LIB=-
L$(TPCC_SQLLIB)/lib -ldb2
#LDFLAGS_LIB=-Xlinker -m -Xlinker
elf64ppc -L$(TPCC_SQLLIB)/lib -ldb2
LDFLAGS_OUT=-o

# Library Configuration
AR=ar
ARFLAGS=-r -v
ARFLAGS_LIB=
ARFLAGS_OUT=

# OS Commands
ERASE=rm -f
ERASEDIR=$(ERASE) -R
MOVE=mv
COPY=cp

# OS File Extensions & Path
Separators
OBJEXT=.o
LIBEXT=.a
BINEXT=
SLASH=/
CMDSEP=;
```

Src.Common/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 - 2005
```

```
## 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.Common
#

include
$(TPCC_ROOT)/Makefile.config

#
################################
################################
##########
# Preprocessor, Compiler and LInker
Flags
#
################################
################################
##########

BND_OPTS =GRANT PUBLIC \
     MESSAGES $*.bnd.msg
PRP_OPTS =BINDFILE \
     OPTLEVEL 1 \
     ISOLATION RR \
     MESSAGES $*.prep.msg \
     LEVEL $(TPCC_VERSION) \
     NOLINEMACRO

INCLUDE =-
I$(TPCC_SQLLIB)/include -
I$(TPCC_ROOT)/include

CFLAGS =  $(CFLAGS_OS)
$(CFLAGS_DEBUG) $(INCLUDE) \
     -DSQLA_NOLINES -
D$(DB2EDITION) -D$(DB2VERSION)
\
     -D$(TPCC_SPTYPE)

UTIL_OBJ =   tpccmisc$(OBJEXT)
tpccdbg$(OBJEXT) tpccctx$(OBJEXT)

#
################################
################################
##########
# User Targets
#
################################
################################
##########

all:  connect $(UTIL_OBJ) disconnect

clean:
   - $(ERASE) *$(OBJEXT) *.bnd
*.msg tpccctx.c

#
################################
################################
##########
# Helper Targets
#
################################
################################
##########

connect:
   - db2 connect to $(TPCC_DBNAME)

disconnect:
   - db2 connect reset
   - db2 terminate

rebind:  connect
   db2 bind tpccctx.bnd $(BND_OPTS)

#
################################
################################
##########
# Build Rules
#
################################
################################
##########

.SUFFIXES:
.SUFFIXES: $(OBJEXT) .c .sqc

.sqc.c:
   @echo "Prepping $*.sqc"
   -db2 prep $*.sqc $(PRP_OPTS)
   @echo "Binding $*.bnd"
   db2 bind $*.bnd $(BND_OPTS)

#
################################
################################
##########
# Dependencies
#
################################
################################
##########

# Source
tpccdbg$(OBJEXT):tpccdbg.c
tpccctx$(OBJEXT):  tpccctx.c
tpccmisc$(OBJEXT):  tpccmisc.c

# Headers
tpccdbg.c:
   $(TPCC_ROOT)/include/db2tpcc.h
```

Src.Common/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 - 2005
** 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 <stdlib.h>
#include <sys/types.h>
#include <sys/time.h>

double current_time_ms(void);
double current_time(void);


/* Current time in SECONDS,
precision SECONDS */
double current_time(void)
{
  /* use time() to get seconds */
  return(time(NULL));
}

/* Current time in SECONDS,
precision MILLISECONDS */
double current_time_ms(void)
{
   /* gettimeofday() returns seconds
and microseconds */
   /* convert to fractional seconds */
  struct timeval t;
  gettimeofday(&t,NULL);
  return (t.tv_sec +
(double)t.tv_usec/(1000*1000));
}
```

dbgen/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 - 2005
## 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) \

   $(TPCC_ROOT)/Src.Common/tpcc
misc$(OBJEXT)
OBJ_EEE =
   $(TPCC_ROOT)/Src.Common/tpccl
wh$(OBJEXT)

EXEC =   gendata$(BINEXT)

#
##############################
##############################
##########
# End-User Targets
#
##############################
##############################
##########

all:  $(EXEC)

clean:
   - $(ERASE) *$(OBJEXT) $(EXEC)

#
##############################
##############################
##########
# Build Rules
#
##############################
##############################
##########

.SUFFIXES:
.SUFFIXES:      $(OBJEXT) .c

# We use $@$(OBJEXT) here so that
the UNIX makefiles work with both
# 'traditional' make and GNU make
$(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
```

dbgen/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 - 2005
** 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;
int quiet_mode = 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|%
d|%d\n";
char fmtItem[]    =
"%s|%d|%s|%d|%d\n";
char fmtStock[]   =
"%d|%d|%d|%d|%s|%s|%s|%s|%s|%s|
%s|%s|%s|%s|%s|%d|%d\n";
char fmtCust[]    =
"%d|%s|%s|%s|%lld|%lld|%s|%s|%d|%
s|%s|%s|%s|%s|%s|%d|%d|%d|%lld|%
lld|%d\n";
char fmtHist[]    =
"%d|%d|%d|%d|%d|%lld|%d|%s\n";
char fmtOrdr[]    =
"%d|%lld|%d|%d|%d|%d|%d|%d\n";
char fmtOLine[]   =
"%lld|%d|%d|%d|%d|%s|%d|%d|%d|%
d\n";
char fmtNewOrd[]  = "%d|%d|%d\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 3 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
   * Quiet Mode:              -q
   *
   * The -f[n] and/or -p[n] options are
required.
   * The -t, -d, -r, -s and -q 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 if (strcmp(argv[i], "-q") == 0)
{
      quiet_mode = 1;
    } 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;
}

/*-----------------------------------------------
--------------------*/
/* generate item table
*/
/*-----------------------------------------------
--------------------*/

void gen_item_tbl( void )
{
   sqlint32  item_num = 0 ;
   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_da
ta, 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 = 0 ;
   sqlint32 stock_num = 0 ;
   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 (!quiet_mode &&
(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_0
1, 24, 24);

create_random_a_string( stock_dist_0
2, 24, 24);

create_random_a_string( stock_dist_0
3, 24, 24);

create_random_a_string( stock_dist_0
4, 24, 24);

create_random_a_string( stock_dist_0
5, 24, 24);

create_random_a_string( stock_dist_0
6, 24, 24);

create_random_a_string( stock_dist_0
7, 24, 24);

create_random_a_string( stock_dist_0
8, 24, 24);

create_random_a_string( stock_dist_0
9, 24, 24);

create_random_a_string( stock_dist_1
0, 24, 24);

         /* create ORIGINAL field */

create_a_string_with_original( stock_d
ata, 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 = 0 ;
   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 ;
   sqlint64 ware_YTD ;

   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++)
   {
      if (!quiet_mode && ((ware_num %
500) == 0)) {              //@dxxxxxxmte
         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);
      ware_YTD = 30000000;

      numBytes = sprintf(Buffer,
fmtWare,
                    ware_name,
                    ware_street_1,
                    ware_street_2,
                    ware_city,
                    ware_state,
                    ware_zip,
                    ware_tax,
                    ware_YTD,
                    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 = 0 ;
   sqlint32 dist_num = 0 ;
   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;
   sqlint64 dist_YTD;

   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++)
   {
      if (!quiet_mode) {
         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);
         dist_YTD = 3000000;

         numBytes = sprintf(Buffer,
fmtDist,
                    next_o_id,
                    dist_tax,
                    dist_YTD,
                    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,ela
pse);
      fflush(stderr);
   }
}

/*-----------------------------------------------
---------------------*/
/* generate customer table
*/
/*-----------------------------------------------
---------------------*/
void gen_cust_tbl( void )
{
   sqlint32 ware_num = 0 ;
   sqlint32 dist_num = 0 ;
   sqlint32 cust_num = 0 ;
   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;
   sqlint64 cust_balance;
   sqlint64 cust_YTD_payment;
   sqlint64 cust_credit_lim;

   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++)
   {
      if (!quiet_mode) {
         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';
            }

            cust_credit_lim = 5000000;
            cust_balance = -1000;
            cust_YTD_payment = 1000;

            numBytes = sprintf(Buffer,
fmtCust,
                        cust_num,
                        cust_state,
                        cust_zip,
                        cust_phone,
                        currtmstmp,
                        cust_credit_lim,
                        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,
                        cust_balance,

cust_YTD_payment,
                        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 = 0 ;
   sqlint32 dist_num = 0 ;
   sqlint32 cust_num = 0 ;
   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++)
   {
      if (!quiet_mode) {
         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 = 0 ;
   sqlint32 dist_num = 0 ;
   sqlint32 nu_ord_id = 0 ;
   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++)
   {
      if (!quiet_mode) {
         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 = 0 ;
   sqlint32 dist_num = 0 ;
   sqlint32 cust_num = 0 ;
   sqlint32 ord_num = 0 ;
   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++)
   {
      if (!quiet_mode) {
```

```
        fprintf(stdout, "ORDERS &
ORDER_LINE for Warehouse #%d\n",
ware_num);
        fflush(stdout);
      }
      for (dist_num = 1; dist_num <=
DISTRICTS_PER_WAREHOUSE;
dist_num++)
      {
         if (!quiet_mode) {
            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_in
fo, 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");
}
```

dbgen/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 - 2005
** 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_DISTRI
CT];

static char alnum[] =

"0123456789abcdefghijklmnopqrstuvw
xyzABCDEFGHIJKLMNOPQRSTUVW
XYZ";

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_integ
er(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,"ORIGIN
AL",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));
}
```

include/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 - 2005
** 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.            */
/* Analysis indicates that a C_LAST
delta of 85 is optimal.              */
/*********************************************
*******************************/
#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 CLIENT_SQL   0
#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

/*
*****************************************
************************** */
/* DB2 Connect/Disconnect & Thread
Context Wrappers                    */
/*
*****************************************
************************** */

#ifdef __cplusplus
extern "C" {
#endif

extern int connect_to_TM(char*);
extern int connect_to_TM_auth(char*,
char*, char*);
extern int disconnect_from_TM(void);


#ifdef __cplusplus
}
#endif

#endif // __DB2TPCC_H
```

include/lval.h

```
#ifndef __LVAL_H
#define __LVAL_H
#define WAREHOUSES 16800
#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
```

include/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 - 2005
** 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
```

include/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 - 2005
** 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
```

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 - 2005
## 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 Kit Version
export TPCC_VERSION=CK041012

# 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}/func
tion
export TPCC_FENCED=NO
```

Appendix - D: RTE Scripts

D.1 RTE Parameters

rteparams

```
/* C-Delta be the difference between C-
LOAD and C-Run.        */
/* C-Delta must be a value between 65..119
including the      */
/* values of 65 and 119 and excluding the
value of 96 and 112 */
#define MASTER_NUM1 0
#define MASTER_NUM2 0
#define MASTER_NUM3 0
#define MASTER_NUM4 0
#define MASTER_NUM5 1
#define MASTER_NUM6 0
#define MASTER_NUM7 0
#define MASTER_NUM8 0
#define MASTER_NUM9 0
#define MASTER_NUM10 0
#define MASTER_NUM11 0
#define MASTER_NUM12 0
#define MASTER_NUM13 0
#define MASTER_NUM14 0
#define MASTER_NUM15 0
#define MASTER_NUM16 0
#define MASTER_NUM17 0
#define MASTER_NUM18 0
#define MASTER_NUM19 0
#define MASTER_NUM20 0
#define MASTER_NUM21 0
#define MASTER_NUM22 0
#define MASTER_NUM23 0
#define MASTER_NUM24 0
#define MASTER_NUM25 0
#define MASTER_NUM26 0
#define MASTER_NUM27 0
#define MASTER_NUM28 0
#define MASTER_NUM29 0
#define MASTER_NUM30 0
#define MASTER_NUM31 0
#define MASTER_NUM32 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"
#elif MASTER_NUM11
MASTER "master11"
#elif MASTER_NUM12
MASTER "master12"
#elif MASTER_NUM13
MASTER "master13"
#elif MASTER_NUM14
MASTER "master14"
#elif MASTER_NUM15
MASTER "master15"
#elif MASTER_NUM16
MASTER "master16"
#elif MASTER_NUM17
MASTER "master17"
#elif MASTER_NUM18
MASTER "master18"
#elif MASTER_NUM19
MASTER "master19"
#elif MASTER_NUM20
MASTER "master20"
#elif MASTER_NUM21
MASTER "master21"
#elif MASTER_NUM22
MASTER "master22"
#elif MASTER_NUM23
MASTER "master23"
#elif MASTER_NUM24
MASTER "master24"
#elif MASTER_NUM25
MASTER "master25"
#elif MASTER_NUM26
MASTER "master26"
#elif MASTER_NUM27
MASTER "master27"
#elif MASTER_NUM28
MASTER "master28"
#elif MASTER_NUM29
MASTER "master29"
#elif MASTER_NUM30
MASTER "master30"
#elif MASTER_NUM31
MASTER "master31"
#elif MASTER_NUM32
MASTER "master32"
#endif
/*---- SUT ----------------------------------------
--*/
SUT="toraus"
/*-------------------------------------------------------
*/
LASTC=88
MEASUREMENT="1"
WAREHOUSES=256000
/*----- SLAVES -----------------------------------
-----*/
#if MASTER_NUM1
SLAVES driver1a, driver1b, driver1c,
driver1d, driver2a, driver2b, driver2c,
driver2d, driver3a, driver3b, driver3c,
driver3d, driver4a, driver4b, driver4c,
driver4d
#elif MASTER_NUM2
SLAVES driver5a, driver5b, driver5c,
driver5d, driver6a, driver6b, driver6c,
driver6d, driver7a, driver7b, driver7c,
driver7d, driver8a, driver8b, driver8c,
driver8d
#elif MASTER_NUM3
SLAVES driver9a, driver9b, driver9c,
driver9d, driver10a, driver10b, driver10c,
driver10d, driver11a, driver11b, driver11c,
driver11d, driver12a, driver12b, driver12c,
driver12d
#elif MASTER_NUM4
SLAVES driver13a, driver13b, driver13c,
driver13d, driver14a, driver14b, driver14c,
driver14d, driver15a, driver15b, driver15c,
driver15d, driver16a, driver16b, driver16c,
driver16d
#elif MASTER_NUM5
SLAVES driver17a, driver17b, driver17c,
driver17d, driver18a, driver18b, driver18c,
driver18d, driver19a, driver19b, driver19c,
driver19d, driver20a, driver20b, driver20c,
driver20d
#elif MASTER_NUM6
SLAVES driver21a, driver21b, driver21c,
driver21d, driver22a, driver22b, driver22c,
driver22d, driver23a, driver23b, driver23c,
driver23d, driver24a, driver24b, driver24c,
driver24d
#elif MASTER_NUM7
SLAVES driver25a, driver25b, driver25c,
driver25d, driver26a, driver26b, driver26c,
driver26d, driver27a, driver27b, driver27c,
driver27d, driver28a, driver28b, driver28c,
driver28d
#elif MASTER_NUM8
SLAVES driver29a, driver29b, driver29c,
driver29d, driver30a, driver30b, driver30c,
driver30d, driver31a, driver31b, driver31c,
driver31d, driver32a, driver32b, driver32c,
driver32d
#elif MASTER_NUM9
SLAVES driver33a, driver33b, driver33c,
driver33d, driver34a, driver34b, driver34c,
driver34d, driver35a, driver35b, driver35c,
driver35d, driver36a, driver36b, driver36c,
driver36d
#elif MASTER_NUM10
SLAVES driver37a, driver37b, driver37c,
driver37d, driver38a, driver38b, driver38c,
driver38d, driver39a, driver39b, driver39c,
driver39d, driver40a, driver40b, driver40c,
driver40d
#elif MASTER_NUM11
SLAVES driver41a, driver41b, driver41c,
driver41d, driver42a, driver42b, driver42c,
driver42d, driver43a, driver43b, driver43c,
driver43d, driver44a, driver44b, driver44c,
driver44d
#elif MASTER_NUM12
SLAVES driver45a, driver45b, driver45c,
driver45d, driver46a, driver46b, driver46c,
driver46d, driver47a, driver47b, driver47c,
driver47d, driver48a, driver48b, driver48c,
driver48d
#elif MASTER_NUM13
SLAVES driver49a, driver49b, driver49c,
driver49d, driver50a, driver50b, driver50c,
driver50d, driver51a, driver51b, driver51c,
driver51d, driver52a, driver52b, driver52c,
driver52d
#elif MASTER_NUM14
SLAVES driver53a, driver53b, driver53c,
driver53d, driver54a, driver54b, driver54c,
driver54d, driver55a, driver55b, driver55c,
driver55d, driver56a, driver56b, driver56c,
driver56d
#elif MASTER_NUM15
SLAVES driver57a, driver57b, driver57c,
driver57d, driver58a, driver58b, driver58c,
driver58d, driver59a, driver59b, driver59c,
driver59d, driver60a, driver60b, driver60c,
driver60d
#elif MASTER_NUM16
SLAVES driver61a, driver61b, driver61c,
driver61d, driver62a, driver62b, driver62c,
driver62d, driver63a, driver63b, driver63c,
driver63d, driver64a, driver64b, driver64c,
driver64d
#elif MASTER_NUM17
SLAVES driver65a, driver65b, driver65c,
driver65d, driver66a, driver66b, driver66c,
driver66d, driver67a, driver67b, driver67c,
driver67d, driver68a, driver68b, driver68c,
driver68d
#elif MASTER_NUM18
SLAVES driver69a, driver69b, driver69c,
driver69d, driver70a, driver70b, driver70c,
driver70d, driver71a, driver71b, driver71c,
driver71d, driver72a, driver72b, driver72c,
driver72d
#elif MASTER_NUM19
SLAVES driver73a, driver73b, driver73c,
driver73d, driver74a, driver74b, driver74c,
driver74d, driver75a, driver75b, driver75c,
driver75d, driver76a, driver76b, driver76c,
driver76d
#elif MASTER_NUM20
SLAVES driver77a, driver77b, driver77c,
driver77d, driver78a, driver78b, driver78c,
driver78d, driver79a, driver79b, driver79c,
driver79d, driver80a, driver80b, driver80c,
driver80d
#elif MASTER_NUM21
```

```
SLAVES driver81a, driver81b, driver81c,
driver81d, driver82a, driver82b, driver82c,
driver82d, driver83a, driver83b, driver83c,
driver83d, driver84a, driver84b, driver84c,
driver84d
#elif MASTER_NUM22
SLAVES driver85a, driver85b, driver85c,
driver85d, driver86a, driver86b, driver86c,
driver86d, driver87a, driver87b, driver87c,
driver87d, driver88a, driver88b, driver88c,
driver88d
#elif MASTER_NUM23
SLAVES driver89a, driver89b, driver89c,
driver89d, driver90a, driver90b, driver90c,
driver90d, driver91a, driver91b, driver91c,
driver91d, driver92a, driver92b, driver92c,
driver92d
#elif MASTER_NUM24
SLAVES driver93a, driver93b, driver93c,
driver93d, driver94a, driver94b, driver94c,
driver94d, driver95a, driver95b, driver95c,
driver95d, driver96a, driver96b, driver96c,
driver96d
#elif MASTER_NUM25
SLAVES driver97a, driver97b, driver97c,
driver97d, driver98a, driver98b, driver98c,
driver98d, driver99a, driver99b, driver99c,
driver99d, driver100a, driver100b,
driver100c, driver100d
#elif MASTER_NUM26
SLAVES driver101a, driver101b, driver101c,
driver101d, driver102a, driver102b,
driver102c, driver102d, driver103a,
driver103b, driver103c, driver103d,
driver104a, driver104b, driver104c,
driver104d
#elif MASTER_NUM27
SLAVES driver105a, driver105b, driver105c,
driver105d, driver106a, driver106b,
driver106c, driver106d, driver107a,
driver107b, driver107c, driver107d,
driver108a, driver108b, driver108c,
driver108d
#elif MASTER_NUM28
SLAVES driver109a, driver109b, driver109c,
driver109d, driver110a, driver110b,
driver110c, driver110d, driver111a,
driver111b, driver111c, driver111d,
driver112a, driver112b, driver112c,
driver112d
#elif MASTER_NUM29
SLAVES driver113a, driver113b, driver113c,
driver113d, driver114a, driver114b,
driver114c, driver114d, driver115a,
driver115b, driver115c, driver115d,
driver116a, driver116b, driver116c,
driver116d
#elif MASTER_NUM30
SLAVES driver117a, driver117b, driver117c,
driver117d, driver118a, driver118b,
driver118c, driver118d, driver119a,
driver119b, driver119c, driver119d,
driver120a, driver120b, driver120c,
driver120d
#elif MASTER_NUM31
SLAVES driver121a, driver121b, driver121c,
driver121d, driver122a, driver122b,
driver122c, driver122d, driver123a,
driver123b, driver123c, driver123d,
driver124a, driver124b, driver124c,
driver124d
#elif MASTER_NUM32
SLAVES driver125a, driver125b, driver125c,
driver125d, driver126a, driver126b,
driver126c, driver126d, driver127a,
driver127b, driver127c, driver127d,
driver128a, driver128b, driver128c,
driver128d
#endif
/* ---- CLIENTS ------------------------------------
------*/
#if MASTER_NUM1
MAIN_CLIENT = client1
CLIENT_REAL = "client1 client2 client3
client4"
#elif MASTER_NUM2
MAIN_CLIENT = client5
CLIENT_REAL = "client5 client6 client7
client8"
#elif MASTER_NUM3
MAIN_CLIENT = client9
CLIENT_REAL = "client9 client10 client11
client12"
#elif MASTER_NUM4
MAIN_CLIENT = client13
CLIENT_REAL = "client13 client14 client15
client16"
#elif MASTER_NUM5
MAIN_CLIENT = client17
CLIENT_REAL = "client17 client18 client19
client20"
#elif MASTER_NUM6
MAIN_CLIENT = client21
CLIENT_REAL = "client21 client22 client23
client24"
#elif MASTER_NUM7
MAIN_CLIENT = client25
CLIENT_REAL = "client25 client26 client27
client28"
#elif MASTER_NUM8
MAIN_CLIENT = client29
CLIENT_REAL = "client29 client30 client31
client32"
#elif MASTER_NUM9
MAIN_CLIENT = client33
CLIENT_REAL = "client33 client34 client35
client36"
#elif MASTER_NUM10
MAIN_CLIENT = client37
CLIENT_REAL = "client37 client38 client39
client40"
#elif MASTER_NUM11
MAIN_CLIENT = client41
CLIENT_REAL = "client41 client42 client43
client44"
#elif MASTER_NUM12
MAIN_CLIENT = client45
CLIENT_REAL = "client45 client46 client47
client48"
#elif MASTER_NUM13
MAIN_CLIENT = client49
CLIENT_REAL = "client49 client50 client51
client52"
#elif MASTER_NUM14
MAIN_CLIENT = client53
CLIENT_REAL = "client53 client54 client55
client56"
#elif MASTER_NUM15
MAIN_CLIENT = client57
CLIENT_REAL = "client57 client58 client59
client60"
#elif MASTER_NUM16
MAIN_CLIENT = client61
CLIENT_REAL = "client61 client62 client63
client64"
#elif MASTER_NUM17
MAIN_CLIENT = client65
CLIENT_REAL = "client65 client66 client67
client68"
#elif MASTER_NUM18
MAIN_CLIENT = client69
CLIENT_REAL = "client69 client70 client71
client72"
#elif MASTER_NUM19
MAIN_CLIENT = client73
CLIENT_REAL = "client73 client74 client75
client76"
#elif MASTER_NUM20
MAIN_CLIENT = client77
CLIENT_REAL = "client77 client78 client79
client80"
#elif MASTER_NUM21
MAIN_CLIENT = client81
CLIENT_REAL = "client81 client82 client83
client84"
#elif MASTER_NUM22
MAIN_CLIENT = client85
CLIENT_REAL = "client85 client86 client87
client88"
#elif MASTER_NUM23
MAIN_CLIENT = client89
CLIENT_REAL = "client89 client90 client91
client92"
#elif MASTER_NUM24
MAIN_CLIENT = client93
CLIENT_REAL = "client93 client94 client95
client96"
#elif MASTER_NUM25
MAIN_CLIENT = client97
CLIENT_REAL = "client97 client98 client99
client100"
#elif MASTER_NUM26
MAIN_CLIENT = client101
CLIENT_REAL = "client101 client102
client103 client104"
#elif MASTER_NUM27
MAIN_CLIENT = client105
CLIENT_REAL = "client105 client106
client107 client108"
#elif MASTER_NUM28
MAIN_CLIENT = client109
CLIENT_REAL = "client109 client110
client111 client112"
#elif MASTER_NUM29
MAIN_CLIENT = client113
CLIENT_REAL = "client113 client114
client115 client116"
#elif MASTER_NUM30
MAIN_CLIENT = client117
CLIENT_REAL = "client117 client118
client119 client120"
#elif MASTER_NUM31
MAIN_CLIENT = client121
CLIENT_REAL = "client121 client122
client123 client124"
#elif MASTER_NUM32
MAIN_CLIENT = client125
CLIENT_REAL = "client125 client126
client127 client128"
#endif
/*---- more client ctuff ----------------------------
------*/
#if MASTER_NUM1
CLIENT client1a tpcc tpcc
CLIENT client2a tpcc tpcc
CLIENT client3a tpcc tpcc
CLIENT client4a tpcc tpcc
#elif MASTER_NUM2
CLIENT client5a tpcc tpcc
CLIENT client6a tpcc tpcc
CLIENT client7a tpcc tpcc
CLIENT client8a tpcc tpcc
#elif MASTER_NUM3
CLIENT client9a tpcc tpcc
CLIENT client10a tpcc tpcc
CLIENT client11a tpcc tpcc
CLIENT client12a tpcc tpcc
#elif MASTER_NUM4
CLIENT client13a tpcc tpcc
CLIENT client14a tpcc tpcc
CLIENT client15a tpcc tpcc
CLIENT client16a tpcc tpcc
#elif MASTER_NUM5
CLIENT client17a tpcc tpcc
CLIENT client18a tpcc tpcc
CLIENT client19a tpcc tpcc
CLIENT client20a tpcc tpcc
#elif MASTER_NUM6
CLIENT client21a tpcc tpcc
CLIENT client22a tpcc tpcc
CLIENT client23a tpcc tpcc
CLIENT client24a tpcc tpcc
#elif MASTER_NUM7
CLIENT client25a tpcc tpcc
CLIENT client26a tpcc tpcc
CLIENT client27a tpcc tpcc
CLIENT client28a tpcc tpcc
#elif MASTER_NUM8
CLIENT client29a tpcc tpcc
```

```
CLIENT client30a tpcc tpcc
CLIENT client31a tpcc tpcc
CLIENT client32a tpcc tpcc
#elif MASTER_NUM9
CLIENT client33a tpcc tpcc
CLIENT client34a tpcc tpcc
CLIENT client35a tpcc tpcc
CLIENT client36a tpcc tpcc
#elif MASTER_NUM10
CLIENT client37a tpcc tpcc
CLIENT client38a tpcc tpcc
CLIENT client39a tpcc tpcc
CLIENT client40a tpcc tpcc
#elif MASTER_NUM11
CLIENT client41a tpcc tpcc
CLIENT client42a tpcc tpcc
CLIENT client43a tpcc tpcc
CLIENT client44a tpcc tpcc
#elif MASTER_NUM12
CLIENT client45a tpcc tpcc
CLIENT client46a tpcc tpcc
CLIENT client47a tpcc tpcc
CLIENT client48a tpcc tpcc
#elif MASTER_NUM13
CLIENT client49a tpcc tpcc
CLIENT client50a tpcc tpcc
CLIENT client51a tpcc tpcc
CLIENT client52a tpcc tpcc
#elif MASTER_NUM14
CLIENT client53a tpcc tpcc
CLIENT client54a tpcc tpcc
CLIENT client55a tpcc tpcc
CLIENT client56a tpcc tpcc
#elif MASTER_NUM15
CLIENT client57a tpcc tpcc
CLIENT client58a tpcc tpcc
CLIENT client59a tpcc tpcc
CLIENT client60a tpcc tpcc
#elif MASTER_NUM16
CLIENT client61a tpcc tpcc
CLIENT client62a tpcc tpcc
CLIENT client63a tpcc tpcc
CLIENT client64a tpcc tpcc
#elif MASTER_NUM17
CLIENT client65a tpcc tpcc
CLIENT client66a tpcc tpcc
CLIENT client67a tpcc tpcc
CLIENT client68a tpcc tpcc
#elif MASTER_NUM18
CLIENT client69a tpcc tpcc
CLIENT client70a tpcc tpcc
CLIENT client71a tpcc tpcc
CLIENT client72a tpcc tpcc
#elif MASTER_NUM19
CLIENT client73a tpcc tpcc
CLIENT client74a tpcc tpcc
CLIENT client75a tpcc tpcc
CLIENT client76a tpcc tpcc
#elif MASTER_NUM20
CLIENT client77a tpcc tpcc
CLIENT client78a tpcc tpcc
CLIENT client79a tpcc tpcc
CLIENT client80a tpcc tpcc
#elif MASTER_NUM21
CLIENT client81a tpcc tpcc
CLIENT client82a tpcc tpcc
CLIENT client83a tpcc tpcc
CLIENT client84a tpcc tpcc
#elif MASTER_NUM22
CLIENT client85a tpcc tpcc
CLIENT client86a tpcc tpcc
CLIENT client87a tpcc tpcc
CLIENT client88a tpcc tpcc
#elif MASTER_NUM23
CLIENT client89a tpcc tpcc
CLIENT client90a tpcc tpcc
CLIENT client91a tpcc tpcc
CLIENT client92a tpcc tpcc
#elif MASTER_NUM24
CLIENT client93a tpcc tpcc
CLIENT client94a tpcc tpcc
CLIENT client95a tpcc tpcc
CLIENT client96a tpcc tpcc
#elif MASTER_NUM25
CLIENT client97a tpcc tpcc
CLIENT client98a tpcc tpcc
CLIENT client99a tpcc tpcc
CLIENT client100a tpcc tpcc
#elif MASTER_NUM26
CLIENT client101a tpcc tpcc
CLIENT client102a tpcc tpcc
CLIENT client103a tpcc tpcc
CLIENT client104a tpcc tpcc
#elif MASTER_NUM27
CLIENT client105a tpcc tpcc
CLIENT client106a tpcc tpcc
CLIENT client107a tpcc tpcc
CLIENT client108a tpcc tpcc
#elif MASTER_NUM28
CLIENT client109a tpcc tpcc
CLIENT client110a tpcc tpcc
CLIENT client111a tpcc tpcc
CLIENT client112a tpcc tpcc
#elif MASTER_NUM29
CLIENT client113a tpcc tpcc
CLIENT client114a tpcc tpcc
CLIENT client115a tpcc tpcc
CLIENT client116a tpcc tpcc
#elif MASTER_NUM30
CLIENT client117a tpcc tpcc
CLIENT client118a tpcc tpcc
CLIENT client119a tpcc tpcc
CLIENT client120a tpcc tpcc
#elif MASTER_NUM31
CLIENT client121a tpcc tpcc
CLIENT client122a tpcc tpcc
CLIENT client123a tpcc tpcc
CLIENT client124a tpcc tpcc
#elif MASTER_NUM32
CLIENT client125a tpcc tpcc
CLIENT client126a tpcc tpcc
CLIENT client127a tpcc tpcc
CLIENT client128a tpcc tpcc
#endif
/*-----------------------------------------------------------
----*/
TELNET telnet 23
SOCKET socket 199703
/* --- Sockets ----------------------------------------
--------*/
#if MASTER_NUM1
SOCKET_NETWORK socket1   80 driver1a
SOCKET_NETWORK socket2   80 driver1b
SOCKET_NETWORK socket3   80 driver1c
SOCKET_NETWORK socket4   80 driver1d
SOCKET_NETWORK socket5   80 driver1a
SOCKET_NETWORK socket6   80 driver1b
SOCKET_NETWORK socket7   80 driver1c
SOCKET_NETWORK socket8   80 driver1d
SOCKET_NETWORK socket9   80 driver1a
SOCKET_NETWORK socket10   80
driver1b
SOCKET_NETWORK socket11   80
driver1c
SOCKET_NETWORK socket12   80
driver1d
SOCKET_NETWORK socket13   80
driver1a
SOCKET_NETWORK socket14   80
driver1b
SOCKET_NETWORK socket15   80
driver1c
SOCKET_NETWORK socket16   80
driver1d
SOCKET_NETWORK socket17   80
driver1a
SOCKET_NETWORK socket18   80
driver1b
SOCKET_NETWORK socket19   80
driver1c
SOCKET_NETWORK socket20   80
driver1d
SOCKET_NETWORK socket21   80
driver1a
SOCKET_NETWORK socket22   80
driver1b
SOCKET_NETWORK socket23   80
driver1c
SOCKET_NETWORK socket24   80
driver1d
SOCKET_NETWORK socket25   80
driver1a
SOCKET_NETWORK socket26   80
driver1b
SOCKET_NETWORK socket27   80
driver1c
SOCKET_NETWORK socket28   80
driver1d
SOCKET_NETWORK socket29   80
driver1a
SOCKET_NETWORK socket30   80
driver1b
SOCKET_NETWORK socket31   80
driver1c
SOCKET_NETWORK socket32   80
driver1d
SOCKET_NETWORK socket33   80
driver2a
SOCKET_NETWORK socket34   80
driver2b
SOCKET_NETWORK socket35   80
driver2c
SOCKET_NETWORK socket36   80
driver2d
SOCKET_NETWORK socket37   80
driver2a
SOCKET_NETWORK socket38   80
driver2b
SOCKET_NETWORK socket39   80
driver2c
SOCKET_NETWORK socket40   80
driver2d
SOCKET_NETWORK socket41   80
driver2a
SOCKET_NETWORK socket42   80
driver2b
SOCKET_NETWORK socket43   80
driver2c
SOCKET_NETWORK socket44   80
driver2d
SOCKET_NETWORK socket45   80
driver2a
SOCKET_NETWORK socket46   80
driver2b
SOCKET_NETWORK socket47   80
driver2c
SOCKET_NETWORK socket48   80
driver2d
SOCKET_NETWORK socket49   80
driver2a
SOCKET_NETWORK socket50   80
driver2b
SOCKET_NETWORK socket51   80
driver2c
SOCKET_NETWORK socket52   80
driver2d
SOCKET_NETWORK socket53   80
driver2a
SOCKET_NETWORK socket54   80
driver2b
SOCKET_NETWORK socket55   80
driver2c
SOCKET_NETWORK socket56   80
driver2d
SOCKET_NETWORK socket57   80
driver2a
SOCKET_NETWORK socket58   80
driver2b
SOCKET_NETWORK socket59   80
driver2c
SOCKET_NETWORK socket60   80
driver2d
SOCKET_NETWORK socket61   80
driver2a
SOCKET_NETWORK socket62   80
driver2b
SOCKET_NETWORK socket63   80
driver2c
```

```
SOCKET_NETWORK socket64   80
driver2d
SOCKET_NETWORK socket65   80
driver3a
SOCKET_NETWORK socket66   80
driver3b
SOCKET_NETWORK socket67   80
driver3c
SOCKET_NETWORK socket68   80
driver3d
SOCKET_NETWORK socket69   80
driver3a
SOCKET_NETWORK socket70   80
driver3b
SOCKET_NETWORK socket71   80
driver3c
SOCKET_NETWORK socket72   80
driver3d
SOCKET_NETWORK socket73   80
driver3a
SOCKET_NETWORK socket74   80
driver3b
SOCKET_NETWORK socket75   80
driver3c
SOCKET_NETWORK socket76   80
driver3d
SOCKET_NETWORK socket77   80
driver3a
SOCKET_NETWORK socket78   80
driver3b
SOCKET_NETWORK socket79   80
driver3c
SOCKET_NETWORK socket80   80
driver3d
SOCKET_NETWORK socket81   80
driver3a
SOCKET_NETWORK socket82   80
driver3b
SOCKET_NETWORK socket83   80
driver3c
SOCKET_NETWORK socket84   80
driver3d
SOCKET_NETWORK socket85   80
driver3a
SOCKET_NETWORK socket86   80
driver3b
SOCKET_NETWORK socket87   80
driver3c
SOCKET_NETWORK socket88   80
driver3d
SOCKET_NETWORK socket89   80
driver3a
SOCKET_NETWORK socket90   80
driver3b
SOCKET_NETWORK socket91   80
driver3c
SOCKET_NETWORK socket92   80
driver3d
SOCKET_NETWORK socket93   80
driver3a
SOCKET_NETWORK socket94   80
driver3b
SOCKET_NETWORK socket95   80
driver3c
SOCKET_NETWORK socket96   80
driver3d
SOCKET_NETWORK socket97   80
driver4a
SOCKET_NETWORK socket98   80
driver4b
SOCKET_NETWORK socket99   80
driver4c
SOCKET_NETWORK socket100   80
driver4d
SOCKET_NETWORK socket101   80
driver4a
SOCKET_NETWORK socket102   80
driver4b
SOCKET_NETWORK socket103   80
driver4c
SOCKET_NETWORK socket104   80
driver4d
SOCKET_NETWORK socket105   80
driver4a
SOCKET_NETWORK socket106   80
driver4b
SOCKET_NETWORK socket107   80
driver4c
SOCKET_NETWORK socket108   80
driver4d
SOCKET_NETWORK socket109   80
driver4a
SOCKET_NETWORK socket110   80
driver4b
SOCKET_NETWORK socket111   80
driver4c
SOCKET_NETWORK socket112   80
driver4d
SOCKET_NETWORK socket113   80
driver4a
SOCKET_NETWORK socket114   80
driver4b
SOCKET_NETWORK socket115   80
driver4c
SOCKET_NETWORK socket116   80
driver4d
SOCKET_NETWORK socket117   80
driver4a
SOCKET_NETWORK socket118   80
driver4b
SOCKET_NETWORK socket119   80
driver4c
SOCKET_NETWORK socket120   80
driver4d
SOCKET_NETWORK socket121   80
driver4a
SOCKET_NETWORK socket122   80
driver4b
SOCKET_NETWORK socket123   80
driver4c
SOCKET_NETWORK socket124   80
driver4d
SOCKET_NETWORK socket125   80
driver4a
SOCKET_NETWORK socket126   80
driver4b
SOCKET_NETWORK socket127   80
driver4c
SOCKET_NETWORK socket128   80
driver4d
#elif MASTER_NUM2
SOCKET_NETWORK socket129   80
driver5a
SOCKET_NETWORK socket130   80
driver5b
SOCKET_NETWORK socket131   80
driver5c
SOCKET_NETWORK socket132   80
driver5d
SOCKET_NETWORK socket133   80
driver5a
SOCKET_NETWORK socket134   80
driver5b
SOCKET_NETWORK socket135   80
driver5c
SOCKET_NETWORK socket136   80
driver5d
SOCKET_NETWORK socket137   80
driver5a
SOCKET_NETWORK socket138   80
driver5b
SOCKET_NETWORK socket139   80
driver5c
SOCKET_NETWORK socket140   80
driver5d
SOCKET_NETWORK socket141   80
driver5a
SOCKET_NETWORK socket142   80
driver5b
SOCKET_NETWORK socket143   80
driver5c
SOCKET_NETWORK socket144   80
driver5d
SOCKET_NETWORK socket145   80
driver5a
SOCKET_NETWORK socket146   80
driver5b
SOCKET_NETWORK socket147   80
driver5c
SOCKET_NETWORK socket148   80
driver5d
SOCKET_NETWORK socket149   80
driver5a
SOCKET_NETWORK socket150   80
driver5b
SOCKET_NETWORK socket151   80
driver5c
SOCKET_NETWORK socket152   80
driver5d
SOCKET_NETWORK socket153   80
driver5a
SOCKET_NETWORK socket154   80
driver5b
SOCKET_NETWORK socket155   80
driver5c
SOCKET_NETWORK socket156   80
driver5d
SOCKET_NETWORK socket157   80
driver5a
SOCKET_NETWORK socket158   80
driver5b
SOCKET_NETWORK socket159   80
driver5c
SOCKET_NETWORK socket160   80
driver5d
SOCKET_NETWORK socket161   80
driver6a
SOCKET_NETWORK socket162   80
driver6b
SOCKET_NETWORK socket163   80
driver6c
SOCKET_NETWORK socket164   80
driver6d
SOCKET_NETWORK socket165   80
driver6a
SOCKET_NETWORK socket166   80
driver6b
SOCKET_NETWORK socket167   80
driver6c
SOCKET_NETWORK socket168   80
driver6d
SOCKET_NETWORK socket169   80
driver6a
SOCKET_NETWORK socket170   80
driver6b
SOCKET_NETWORK socket171   80
driver6c
SOCKET_NETWORK socket172   80
driver6d
SOCKET_NETWORK socket173   80
driver6a
SOCKET_NETWORK socket174   80
driver6b
SOCKET_NETWORK socket175   80
driver6c
SOCKET_NETWORK socket176   80
driver6d
SOCKET_NETWORK socket177   80
driver6a
SOCKET_NETWORK socket178   80
driver6b
SOCKET_NETWORK socket179   80
driver6c
SOCKET_NETWORK socket180   80
driver6d
SOCKET_NETWORK socket181   80
driver6a
SOCKET_NETWORK socket182   80
driver6b
SOCKET_NETWORK socket183   80
driver6c
SOCKET_NETWORK socket184   80
driver6d
SOCKET_NETWORK socket185   80
driver6a
SOCKET_NETWORK socket186   80
driver6b
```

```
SOCKET_NETWORK socket187  80
driver6c
SOCKET_NETWORK socket188  80
driver6d
SOCKET_NETWORK socket189  80
driver6a
SOCKET_NETWORK socket190  80
driver6b
SOCKET_NETWORK socket191  80
driver6c
SOCKET_NETWORK socket192  80
driver6d
SOCKET_NETWORK socket193  80
driver7a
SOCKET_NETWORK socket194  80
driver7b
SOCKET_NETWORK socket195  80
driver7c
SOCKET_NETWORK socket196  80
driver7d
SOCKET_NETWORK socket197  80
driver7a
SOCKET_NETWORK socket198  80
driver7b
SOCKET_NETWORK socket199  80
driver7c
SOCKET_NETWORK socket200  80
driver7d
SOCKET_NETWORK socket201  80
driver7a
SOCKET_NETWORK socket202  80
driver7b
SOCKET_NETWORK socket203  80
driver7c
SOCKET_NETWORK socket204  80
driver7d
SOCKET_NETWORK socket205  80
driver7a
SOCKET_NETWORK socket206  80
driver7b
SOCKET_NETWORK socket207  80
driver7c
SOCKET_NETWORK socket208  80
driver7d
SOCKET_NETWORK socket209  80
driver7a
SOCKET_NETWORK socket210  80
driver7b
SOCKET_NETWORK socket211  80
driver7c
SOCKET_NETWORK socket212  80
driver7d
SOCKET_NETWORK socket213  80
driver7a
SOCKET_NETWORK socket214  80
driver7b
SOCKET_NETWORK socket215  80
driver7c
SOCKET_NETWORK socket216  80
driver7d
SOCKET_NETWORK socket217  80
driver7a
SOCKET_NETWORK socket218  80
driver7b
SOCKET_NETWORK socket219  80
driver7c
SOCKET_NETWORK socket220  80
driver7d
SOCKET_NETWORK socket221  80
driver7a
SOCKET_NETWORK socket222  80
driver7b
SOCKET_NETWORK socket223  80
driver7c
SOCKET_NETWORK socket224  80
driver7d
SOCKET_NETWORK socket225  80
driver8a
SOCKET_NETWORK socket226  80
driver8b
SOCKET_NETWORK socket227  80
driver8c
SOCKET_NETWORK socket228  80
driver8d
SOCKET_NETWORK socket229  80
driver8a
SOCKET_NETWORK socket230  80
driver8b
SOCKET_NETWORK socket231  80
driver8c
SOCKET_NETWORK socket232  80
driver8d
SOCKET_NETWORK socket233  80
driver8a
SOCKET_NETWORK socket234  80
driver8b
SOCKET_NETWORK socket235  80
driver8c
SOCKET_NETWORK socket236  80
driver8d
SOCKET_NETWORK socket237  80
driver8a
SOCKET_NETWORK socket238  80
driver8b
SOCKET_NETWORK socket239  80
driver8c
SOCKET_NETWORK socket240  80
driver8d
SOCKET_NETWORK socket241  80
driver8a
SOCKET_NETWORK socket242  80
driver8b
SOCKET_NETWORK socket243  80
driver8c
SOCKET_NETWORK socket244  80
driver8d
SOCKET_NETWORK socket245  80
driver8a
SOCKET_NETWORK socket246  80
driver8b
SOCKET_NETWORK socket247  80
driver8c
SOCKET_NETWORK socket248  80
driver8d
SOCKET_NETWORK socket249  80
driver8a
SOCKET_NETWORK socket250  80
driver8b
SOCKET_NETWORK socket251  80
driver8c
SOCKET_NETWORK socket252  80
driver8d
SOCKET_NETWORK socket253  80
driver8a
SOCKET_NETWORK socket254  80
driver8b
SOCKET_NETWORK socket255  80
driver8c
SOCKET_NETWORK socket256  80
driver8d
#elif MASTER_NUM3
SOCKET_NETWORK socket257  80
driver9a
SOCKET_NETWORK socket258  80
driver9b
SOCKET_NETWORK socket259  80
driver9c
SOCKET_NETWORK socket260  80
driver9d
SOCKET_NETWORK socket261  80
driver9a
SOCKET_NETWORK socket262  80
driver9b
SOCKET_NETWORK socket263  80
driver9c
SOCKET_NETWORK socket264  80
driver9d
SOCKET_NETWORK socket265  80
driver9a
SOCKET_NETWORK socket266  80
driver9b
SOCKET_NETWORK socket267  80
driver9c
SOCKET_NETWORK socket268  80
driver9d
SOCKET_NETWORK socket269  80
driver9a
SOCKET_NETWORK socket270  80
driver9b
SOCKET_NETWORK socket271  80
driver9c
SOCKET_NETWORK socket272  80
driver9d
SOCKET_NETWORK socket273  80
driver9a
SOCKET_NETWORK socket274  80
driver9b
SOCKET_NETWORK socket275  80
driver9c
SOCKET_NETWORK socket276  80
driver9d
SOCKET_NETWORK socket277  80
driver9a
SOCKET_NETWORK socket278  80
driver9b
SOCKET_NETWORK socket279  80
driver9c
SOCKET_NETWORK socket280  80
driver9d
SOCKET_NETWORK socket281  80
driver9a
SOCKET_NETWORK socket282  80
driver9b
SOCKET_NETWORK socket283  80
driver9c
SOCKET_NETWORK socket284  80
driver9d
SOCKET_NETWORK socket285  80
driver9a
SOCKET_NETWORK socket286  80
driver9b
SOCKET_NETWORK socket287  80
driver9c
SOCKET_NETWORK socket288  80
driver9d
SOCKET_NETWORK socket289  80
driver10a
SOCKET_NETWORK socket290  80
driver10b
SOCKET_NETWORK socket291  80
driver10c
SOCKET_NETWORK socket292  80
driver10d
SOCKET_NETWORK socket293  80
driver10a
SOCKET_NETWORK socket294  80
driver10b
SOCKET_NETWORK socket295  80
driver10c
SOCKET_NETWORK socket296  80
driver10d
SOCKET_NETWORK socket297  80
driver10a
SOCKET_NETWORK socket298  80
driver10b
SOCKET_NETWORK socket299  80
driver10c
SOCKET_NETWORK socket300  80
driver10d
SOCKET_NETWORK socket301  80
driver10a
SOCKET_NETWORK socket302  80
driver10b
SOCKET_NETWORK socket303  80
driver10c
SOCKET_NETWORK socket304  80
driver10d
SOCKET_NETWORK socket305  80
driver10a
SOCKET_NETWORK socket306  80
driver10b
SOCKET_NETWORK socket307  80
driver10c
SOCKET_NETWORK socket308  80
driver10d
SOCKET_NETWORK socket309  80
driver10a
```

```
SOCKET_NETWORK socket310  80
driver10b
SOCKET_NETWORK socket311  80
driver10c
SOCKET_NETWORK socket312  80
driver10d
SOCKET_NETWORK socket313  80
driver10a
SOCKET_NETWORK socket314  80
driver10b
SOCKET_NETWORK socket315  80
driver10c
SOCKET_NETWORK socket316  80
driver10d
SOCKET_NETWORK socket317  80
driver10a
SOCKET_NETWORK socket318  80
driver10b
SOCKET_NETWORK socket319  80
driver10c
SOCKET_NETWORK socket320  80
driver10d
SOCKET_NETWORK socket321  80
driver11a
SOCKET_NETWORK socket322  80
driver11b
SOCKET_NETWORK socket323  80
driver11c
SOCKET_NETWORK socket324  80
driver11d
SOCKET_NETWORK socket325  80
driver11a
SOCKET_NETWORK socket326  80
driver11b
SOCKET_NETWORK socket327  80
driver11c
SOCKET_NETWORK socket328  80
driver11d
SOCKET_NETWORK socket329  80
driver11a
SOCKET_NETWORK socket330  80
driver11b
SOCKET_NETWORK socket331  80
driver11c
SOCKET_NETWORK socket332  80
driver11d
SOCKET_NETWORK socket333  80
driver11a
SOCKET_NETWORK socket334  80
driver11b
SOCKET_NETWORK socket335  80
driver11c
SOCKET_NETWORK socket336  80
driver11d
SOCKET_NETWORK socket337  80
driver11a
SOCKET_NETWORK socket338  80
driver11b
SOCKET_NETWORK socket339  80
driver11c
SOCKET_NETWORK socket340  80
driver11d
SOCKET_NETWORK socket341  80
driver11a
SOCKET_NETWORK socket342  80
driver11b
SOCKET_NETWORK socket343  80
driver11c
SOCKET_NETWORK socket344  80
driver11d
SOCKET_NETWORK socket345  80
driver11a
SOCKET_NETWORK socket346  80
driver11b
SOCKET_NETWORK socket347  80
driver11c
SOCKET_NETWORK socket348  80
driver11d
SOCKET_NETWORK socket349  80
driver11a
SOCKET_NETWORK socket350  80
driver11b
SOCKET_NETWORK socket351  80
driver11c
SOCKET_NETWORK socket352  80
driver11d
SOCKET_NETWORK socket353  80
driver12a
SOCKET_NETWORK socket354  80
driver12b
SOCKET_NETWORK socket355  80
driver12c
SOCKET_NETWORK socket356  80
driver12d
SOCKET_NETWORK socket357  80
driver12a
SOCKET_NETWORK socket358  80
driver12b
SOCKET_NETWORK socket359  80
driver12c
SOCKET_NETWORK socket360  80
driver12d
SOCKET_NETWORK socket361  80
driver12a
SOCKET_NETWORK socket362  80
driver12b
SOCKET_NETWORK socket363  80
driver12c
SOCKET_NETWORK socket364  80
driver12d
SOCKET_NETWORK socket365  80
driver12a
SOCKET_NETWORK socket366  80
driver12b
SOCKET_NETWORK socket367  80
driver12c
SOCKET_NETWORK socket368  80
driver12d
SOCKET_NETWORK socket369  80
driver12a
SOCKET_NETWORK socket370  80
driver12b
SOCKET_NETWORK socket371  80
driver12c
SOCKET_NETWORK socket372  80
driver12d
SOCKET_NETWORK socket373  80
driver12a
SOCKET_NETWORK socket374  80
driver12b
SOCKET_NETWORK socket375  80
driver12c
SOCKET_NETWORK socket376  80
driver12d
SOCKET_NETWORK socket377  80
driver12a
SOCKET_NETWORK socket378  80
driver12b
SOCKET_NETWORK socket379  80
driver12c
SOCKET_NETWORK socket380  80
driver12d
SOCKET_NETWORK socket381  80
driver12a
SOCKET_NETWORK socket382  80
driver12b
SOCKET_NETWORK socket383  80
driver12c
SOCKET_NETWORK socket384  80
driver12d
#elif MASTER_NUM4
SOCKET_NETWORK socket385  80
driver13a
SOCKET_NETWORK socket386  80
driver13b
SOCKET_NETWORK socket387  80
driver13c
SOCKET_NETWORK socket388  80
driver13d
SOCKET_NETWORK socket389  80
driver13a
SOCKET_NETWORK socket390  80
driver13b
SOCKET_NETWORK socket391  80
driver13c
SOCKET_NETWORK socket392  80
driver13d
SOCKET_NETWORK socket393  80
driver13a
SOCKET_NETWORK socket394  80
driver13b
SOCKET_NETWORK socket395  80
driver13c
SOCKET_NETWORK socket396  80
driver13d
SOCKET_NETWORK socket397  80
driver13a
SOCKET_NETWORK socket398  80
driver13b
SOCKET_NETWORK socket399  80
driver13c
SOCKET_NETWORK socket400  80
driver13d
SOCKET_NETWORK socket401  80
driver13a
SOCKET_NETWORK socket402  80
driver13b
SOCKET_NETWORK socket403  80
driver13c
SOCKET_NETWORK socket404  80
driver13d
SOCKET_NETWORK socket405  80
driver13a
SOCKET_NETWORK socket406  80
driver13b
SOCKET_NETWORK socket407  80
driver13c
SOCKET_NETWORK socket408  80
driver13d
SOCKET_NETWORK socket409  80
driver13a
SOCKET_NETWORK socket410  80
driver13b
SOCKET_NETWORK socket411  80
driver13c
SOCKET_NETWORK socket412  80
driver13d
SOCKET_NETWORK socket413  80
driver13a
SOCKET_NETWORK socket414  80
driver13b
SOCKET_NETWORK socket415  80
driver13c
SOCKET_NETWORK socket416  80
driver13d
SOCKET_NETWORK socket417  80
driver14a
SOCKET_NETWORK socket418  80
driver14b
SOCKET_NETWORK socket419  80
driver14c
SOCKET_NETWORK socket420  80
driver14d
SOCKET_NETWORK socket421  80
driver14a
SOCKET_NETWORK socket422  80
driver14b
SOCKET_NETWORK socket423  80
driver14c
SOCKET_NETWORK socket424  80
driver14d
SOCKET_NETWORK socket425  80
driver14a
SOCKET_NETWORK socket426  80
driver14b
SOCKET_NETWORK socket427  80
driver14c
SOCKET_NETWORK socket428  80
driver14d
SOCKET_NETWORK socket429  80
driver14a
SOCKET_NETWORK socket430  80
driver14b
SOCKET_NETWORK socket431  80
driver14c
SOCKET_NETWORK socket432  80
driver14d
```

```
SOCKET_NETWORK socket433  80
driver14a
SOCKET_NETWORK socket434  80
driver14b
SOCKET_NETWORK socket435  80
driver14c
SOCKET_NETWORK socket436  80
driver14d
SOCKET_NETWORK socket437  80
driver14a
SOCKET_NETWORK socket438  80
driver14b
SOCKET_NETWORK socket439  80
driver14c
SOCKET_NETWORK socket440  80
driver14d
SOCKET_NETWORK socket441  80
driver14a
SOCKET_NETWORK socket442  80
driver14b
SOCKET_NETWORK socket443  80
driver14c
SOCKET_NETWORK socket444  80
driver14d
SOCKET_NETWORK socket445  80
driver14a
SOCKET_NETWORK socket446  80
driver14b
SOCKET_NETWORK socket447  80
driver14c
SOCKET_NETWORK socket448  80
driver14d
SOCKET_NETWORK socket449  80
driver15a
SOCKET_NETWORK socket450  80
driver15b
SOCKET_NETWORK socket451  80
driver15c
SOCKET_NETWORK socket452  80
driver15d
SOCKET_NETWORK socket453  80
driver15a
SOCKET_NETWORK socket454  80
driver15b
SOCKET_NETWORK socket455  80
driver15c
SOCKET_NETWORK socket456  80
driver15d
SOCKET_NETWORK socket457  80
driver15a
SOCKET_NETWORK socket458  80
driver15b
SOCKET_NETWORK socket459  80
driver15c
SOCKET_NETWORK socket460  80
driver15d
SOCKET_NETWORK socket461  80
driver15a
SOCKET_NETWORK socket462  80
driver15b
SOCKET_NETWORK socket463  80
driver15c
SOCKET_NETWORK socket464  80
driver15d
SOCKET_NETWORK socket465  80
driver15a
SOCKET_NETWORK socket466  80
driver15b
SOCKET_NETWORK socket467  80
driver15c
SOCKET_NETWORK socket468  80
driver15d
SOCKET_NETWORK socket469  80
driver15a
SOCKET_NETWORK socket470  80
driver15b
SOCKET_NETWORK socket471  80
driver15c
SOCKET_NETWORK socket472  80
driver15d
SOCKET_NETWORK socket473  80
driver15a
SOCKET_NETWORK socket474  80
driver15b
SOCKET_NETWORK socket475  80
driver15c
SOCKET_NETWORK socket476  80
driver15d
SOCKET_NETWORK socket477  80
driver15a
SOCKET_NETWORK socket478  80
driver15b
SOCKET_NETWORK socket479  80
driver15c
SOCKET_NETWORK socket480  80
driver15d
SOCKET_NETWORK socket481  80
driver16a
SOCKET_NETWORK socket482  80
driver16b
SOCKET_NETWORK socket483  80
driver16c
SOCKET_NETWORK socket484  80
driver16d
SOCKET_NETWORK socket485  80
driver16a
SOCKET_NETWORK socket486  80
driver16b
SOCKET_NETWORK socket487  80
driver16c
SOCKET_NETWORK socket488  80
driver16d
SOCKET_NETWORK socket489  80
driver16a
SOCKET_NETWORK socket490  80
driver16b
SOCKET_NETWORK socket491  80
driver16c
SOCKET_NETWORK socket492  80
driver16d
SOCKET_NETWORK socket493  80
driver16a
SOCKET_NETWORK socket494  80
driver16b
SOCKET_NETWORK socket495  80
driver16c
SOCKET_NETWORK socket496  80
driver16d
SOCKET_NETWORK socket497  80
driver16a
SOCKET_NETWORK socket498  80
driver16b
SOCKET_NETWORK socket499  80
driver16c
SOCKET_NETWORK socket500  80
driver16d
SOCKET_NETWORK socket501  80
driver16a
SOCKET_NETWORK socket502  80
driver16b
SOCKET_NETWORK socket503  80
driver16c
SOCKET_NETWORK socket504  80
driver16d
SOCKET_NETWORK socket505  80
driver16a
SOCKET_NETWORK socket506  80
driver16b
SOCKET_NETWORK socket507  80
driver16c
SOCKET_NETWORK socket508  80
driver16d
SOCKET_NETWORK socket509  80
driver16a
SOCKET_NETWORK socket510  80
driver16b
SOCKET_NETWORK socket511  80
driver16c
SOCKET_NETWORK socket512  80
driver16d
#elif MASTER_NUM5
SOCKET_NETWORK socket513  80
driver17a
SOCKET_NETWORK socket514  80
driver17b
SOCKET_NETWORK socket515  80
driver17c
SOCKET_NETWORK socket516  80
driver17d
SOCKET_NETWORK socket517  80
driver17a
SOCKET_NETWORK socket518  80
driver17b
SOCKET_NETWORK socket519  80
driver17c
SOCKET_NETWORK socket520  80
driver17d
SOCKET_NETWORK socket521  80
driver17a
SOCKET_NETWORK socket522  80
driver17b
SOCKET_NETWORK socket523  80
driver17c
SOCKET_NETWORK socket524  80
driver17d
SOCKET_NETWORK socket525  80
driver17a
SOCKET_NETWORK socket526  80
driver17b
SOCKET_NETWORK socket527  80
driver17c
SOCKET_NETWORK socket528  80
driver17d
SOCKET_NETWORK socket529  80
driver17a
SOCKET_NETWORK socket530  80
driver17b
SOCKET_NETWORK socket531  80
driver17c
SOCKET_NETWORK socket532  80
driver17d
SOCKET_NETWORK socket533  80
driver17a
SOCKET_NETWORK socket534  80
driver17b
SOCKET_NETWORK socket535  80
driver17c
SOCKET_NETWORK socket536  80
driver17d
SOCKET_NETWORK socket537  80
driver17a
SOCKET_NETWORK socket538  80
driver17b
SOCKET_NETWORK socket539  80
driver17c
SOCKET_NETWORK socket540  80
driver17d
SOCKET_NETWORK socket541  80
driver17a
SOCKET_NETWORK socket542  80
driver17b
SOCKET_NETWORK socket543  80
driver17c
SOCKET_NETWORK socket544  80
driver17d
SOCKET_NETWORK socket545  80
driver18a
SOCKET_NETWORK socket546  80
driver18b
SOCKET_NETWORK socket547  80
driver18c
SOCKET_NETWORK socket548  80
driver18d
SOCKET_NETWORK socket549  80
driver18a
SOCKET_NETWORK socket550  80
driver18b
SOCKET_NETWORK socket551  80
driver18c
SOCKET_NETWORK socket552  80
driver18d
SOCKET_NETWORK socket553  80
driver18a
SOCKET_NETWORK socket554  80
driver18b
SOCKET_NETWORK socket555  80
driver18c
```

```
SOCKET_NETWORK socket556  80
driver18d
SOCKET_NETWORK socket557  80
driver18a
SOCKET_NETWORK socket558  80
driver18b
SOCKET_NETWORK socket559  80
driver18c
SOCKET_NETWORK socket560  80
driver18d
SOCKET_NETWORK socket561  80
driver18a
SOCKET_NETWORK socket562  80
driver18b
SOCKET_NETWORK socket563  80
driver18c
SOCKET_NETWORK socket564  80
driver18d
SOCKET_NETWORK socket565  80
driver18a
SOCKET_NETWORK socket566  80
driver18b
SOCKET_NETWORK socket567  80
driver18c
SOCKET_NETWORK socket568  80
driver18d
SOCKET_NETWORK socket569  80
driver18a
SOCKET_NETWORK socket570  80
driver18b
SOCKET_NETWORK socket571  80
driver18c
SOCKET_NETWORK socket572  80
driver18d
SOCKET_NETWORK socket573  80
driver18a
SOCKET_NETWORK socket574  80
driver18b
SOCKET_NETWORK socket575  80
driver18c
SOCKET_NETWORK socket576  80
driver18d
SOCKET_NETWORK socket577  80
driver19a
SOCKET_NETWORK socket578  80
driver19b
SOCKET_NETWORK socket579  80
driver19c
SOCKET_NETWORK socket580  80
driver19d
SOCKET_NETWORK socket581  80
driver19a
SOCKET_NETWORK socket582  80
driver19b
SOCKET_NETWORK socket583  80
driver19c
SOCKET_NETWORK socket584  80
driver19d
SOCKET_NETWORK socket585  80
driver19a
SOCKET_NETWORK socket586  80
driver19b
SOCKET_NETWORK socket587  80
driver19c
SOCKET_NETWORK socket588  80
driver19d
SOCKET_NETWORK socket589  80
driver19a
SOCKET_NETWORK socket590  80
driver19b
SOCKET_NETWORK socket591  80
driver19c
SOCKET_NETWORK socket592  80
driver19d
SOCKET_NETWORK socket593  80
driver19a
SOCKET_NETWORK socket594  80
driver19b
SOCKET_NETWORK socket595  80
driver19c
SOCKET_NETWORK socket596  80
driver19d
SOCKET_NETWORK socket597  80
driver19a
SOCKET_NETWORK socket598  80
driver19b
SOCKET_NETWORK socket599  80
driver19c
SOCKET_NETWORK socket600  80
driver19d
SOCKET_NETWORK socket601  80
driver19a
SOCKET_NETWORK socket602  80
driver19b
SOCKET_NETWORK socket603  80
driver19c
SOCKET_NETWORK socket604  80
driver19d
SOCKET_NETWORK socket605  80
driver19a
SOCKET_NETWORK socket606  80
driver19b
SOCKET_NETWORK socket607  80
driver19c
SOCKET_NETWORK socket608  80
driver19d
SOCKET_NETWORK socket609  80
driver20a
SOCKET_NETWORK socket610  80
driver20b
SOCKET_NETWORK socket611  80
driver20c
SOCKET_NETWORK socket612  80
driver20d
SOCKET_NETWORK socket613  80
driver20a
SOCKET_NETWORK socket614  80
driver20b
SOCKET_NETWORK socket615  80
driver20c
SOCKET_NETWORK socket616  80
driver20d
SOCKET_NETWORK socket617  80
driver20a
SOCKET_NETWORK socket618  80
driver20b
SOCKET_NETWORK socket619  80
driver20c
SOCKET_NETWORK socket620  80
driver20d
SOCKET_NETWORK socket621  80
driver20a
SOCKET_NETWORK socket622  80
driver20b
SOCKET_NETWORK socket623  80
driver20c
SOCKET_NETWORK socket624  80
driver20d
SOCKET_NETWORK socket625  80
driver20a
SOCKET_NETWORK socket626  80
driver20b
SOCKET_NETWORK socket627  80
driver20c
SOCKET_NETWORK socket628  80
driver20d
SOCKET_NETWORK socket629  80
driver20a
SOCKET_NETWORK socket630  80
driver20b
SOCKET_NETWORK socket631  80
driver20c
SOCKET_NETWORK socket632  80
driver20d
SOCKET_NETWORK socket633  80
driver20a
SOCKET_NETWORK socket634  80
driver20b
SOCKET_NETWORK socket635  80
driver20c
SOCKET_NETWORK socket636  80
driver20d
SOCKET_NETWORK socket637  80
driver20a
SOCKET_NETWORK socket638  80
driver20b
SOCKET_NETWORK socket639  80
driver20c
SOCKET_NETWORK socket640  80
driver20d
#elif MASTER_NUM6
SOCKET_NETWORK socket641  80
driver21a
SOCKET_NETWORK socket642  80
driver21b
SOCKET_NETWORK socket643  80
driver21c
SOCKET_NETWORK socket644  80
driver21d
SOCKET_NETWORK socket645  80
driver21a
SOCKET_NETWORK socket646  80
driver21b
SOCKET_NETWORK socket647  80
driver21c
SOCKET_NETWORK socket648  80
driver21d
SOCKET_NETWORK socket649  80
driver21a
SOCKET_NETWORK socket650  80
driver21b
SOCKET_NETWORK socket651  80
driver21c
SOCKET_NETWORK socket652  80
driver21d
SOCKET_NETWORK socket653  80
driver21a
SOCKET_NETWORK socket654  80
driver21b
SOCKET_NETWORK socket655  80
driver21c
SOCKET_NETWORK socket656  80
driver21d
SOCKET_NETWORK socket657  80
driver21a
SOCKET_NETWORK socket658  80
driver21b
SOCKET_NETWORK socket659  80
driver21c
SOCKET_NETWORK socket660  80
driver21d
SOCKET_NETWORK socket661  80
driver21a
SOCKET_NETWORK socket662  80
driver21b
SOCKET_NETWORK socket663  80
driver21c
SOCKET_NETWORK socket664  80
driver21d
SOCKET_NETWORK socket665  80
driver21a
SOCKET_NETWORK socket666  80
driver21b
SOCKET_NETWORK socket667  80
driver21c
SOCKET_NETWORK socket668  80
driver21d
SOCKET_NETWORK socket669  80
driver21a
SOCKET_NETWORK socket670  80
driver21b
SOCKET_NETWORK socket671  80
driver21c
SOCKET_NETWORK socket672  80
driver21d
SOCKET_NETWORK socket673  80
driver22a
SOCKET_NETWORK socket674  80
driver22b
SOCKET_NETWORK socket675  80
driver22c
SOCKET_NETWORK socket676  80
driver22d
SOCKET_NETWORK socket677  80
driver22a
SOCKET_NETWORK socket678  80
driver22b
```

```
SOCKET_NETWORK socket679   80
driver22c
SOCKET_NETWORK socket680   80
driver22d
SOCKET_NETWORK socket681   80
driver22a
SOCKET_NETWORK socket682   80
driver22b
SOCKET_NETWORK socket683   80
driver22c
SOCKET_NETWORK socket684   80
driver22d
SOCKET_NETWORK socket685   80
driver22a
SOCKET_NETWORK socket686   80
driver22b
SOCKET_NETWORK socket687   80
driver22c
SOCKET_NETWORK socket688   80
driver22d
SOCKET_NETWORK socket689   80
driver22a
SOCKET_NETWORK socket690   80
driver22b
SOCKET_NETWORK socket691   80
driver22c
SOCKET_NETWORK socket692   80
driver22d
SOCKET_NETWORK socket693   80
driver22a
SOCKET_NETWORK socket694   80
driver22b
SOCKET_NETWORK socket695   80
driver22c
SOCKET_NETWORK socket696   80
driver22d
SOCKET_NETWORK socket697   80
driver22a
SOCKET_NETWORK socket698   80
driver22b
SOCKET_NETWORK socket699   80
driver22c
SOCKET_NETWORK socket700   80
driver22d
SOCKET_NETWORK socket701   80
driver22a
SOCKET_NETWORK socket702   80
driver22b
SOCKET_NETWORK socket703   80
driver22c
SOCKET_NETWORK socket704   80
driver22d
SOCKET_NETWORK socket705   80
driver23a
SOCKET_NETWORK socket706   80
driver23b
SOCKET_NETWORK socket707   80
driver23c
SOCKET_NETWORK socket708   80
driver23d
SOCKET_NETWORK socket709   80
driver23a
SOCKET_NETWORK socket710   80
driver23b
SOCKET_NETWORK socket711   80
driver23c
SOCKET_NETWORK socket712   80
driver23d
SOCKET_NETWORK socket713   80
driver23a
SOCKET_NETWORK socket714   80
driver23b
SOCKET_NETWORK socket715   80
driver23c
SOCKET_NETWORK socket716   80
driver23d
SOCKET_NETWORK socket717   80
driver23a
SOCKET_NETWORK socket718   80
driver23b
SOCKET_NETWORK socket719   80
driver23c
SOCKET_NETWORK socket720   80
driver23d
SOCKET_NETWORK socket721   80
driver23a
SOCKET_NETWORK socket722   80
driver23b
SOCKET_NETWORK socket723   80
driver23c
SOCKET_NETWORK socket724   80
driver23d
SOCKET_NETWORK socket725   80
driver23a
SOCKET_NETWORK socket726   80
driver23b
SOCKET_NETWORK socket727   80
driver23c
SOCKET_NETWORK socket728   80
driver23d
SOCKET_NETWORK socket729   80
driver23a
SOCKET_NETWORK socket730   80
driver23b
SOCKET_NETWORK socket731   80
driver23c
SOCKET_NETWORK socket732   80
driver23d
SOCKET_NETWORK socket733   80
driver23a
SOCKET_NETWORK socket734   80
driver23b
SOCKET_NETWORK socket735   80
driver23c
SOCKET_NETWORK socket736   80
driver23d
SOCKET_NETWORK socket737   80
driver24a
SOCKET_NETWORK socket738   80
driver24b
SOCKET_NETWORK socket739   80
driver24c
SOCKET_NETWORK socket740   80
driver24d
SOCKET_NETWORK socket741   80
driver24a
SOCKET_NETWORK socket742   80
driver24b
SOCKET_NETWORK socket743   80
driver24c
SOCKET_NETWORK socket744   80
driver24d
SOCKET_NETWORK socket745   80
driver24a
SOCKET_NETWORK socket746   80
driver24b
SOCKET_NETWORK socket747   80
driver24c
SOCKET_NETWORK socket748   80
driver24d
SOCKET_NETWORK socket749   80
driver24a
SOCKET_NETWORK socket750   80
driver24b
SOCKET_NETWORK socket751   80
driver24c
SOCKET_NETWORK socket752   80
driver24d
SOCKET_NETWORK socket753   80
driver24a
SOCKET_NETWORK socket754   80
driver24b
SOCKET_NETWORK socket755   80
driver24c
SOCKET_NETWORK socket756   80
driver24d
SOCKET_NETWORK socket757   80
driver24a
SOCKET_NETWORK socket758   80
driver24b
SOCKET_NETWORK socket759   80
driver24c
SOCKET_NETWORK socket760   80
driver24d
SOCKET_NETWORK socket761   80
driver24a
SOCKET_NETWORK socket762   80
driver24b
SOCKET_NETWORK socket763   80
driver24c
SOCKET_NETWORK socket764   80
driver24d
SOCKET_NETWORK socket765   80
driver24a
SOCKET_NETWORK socket766   80
driver24b
SOCKET_NETWORK socket767   80
driver24c
SOCKET_NETWORK socket768   80
driver24d
#elif MASTER_NUM7
SOCKET_NETWORK socket769   80
driver25a
SOCKET_NETWORK socket770   80
driver25b
SOCKET_NETWORK socket771   80
driver25c
SOCKET_NETWORK socket772   80
driver25d
SOCKET_NETWORK socket773   80
driver25a
SOCKET_NETWORK socket774   80
driver25b
SOCKET_NETWORK socket775   80
driver25c
SOCKET_NETWORK socket776   80
driver25d
SOCKET_NETWORK socket777   80
driver25a
SOCKET_NETWORK socket778   80
driver25b
SOCKET_NETWORK socket779   80
driver25c
SOCKET_NETWORK socket780   80
driver25d
SOCKET_NETWORK socket781   80
driver25a
SOCKET_NETWORK socket782   80
driver25b
SOCKET_NETWORK socket783   80
driver25c
SOCKET_NETWORK socket784   80
driver25d
SOCKET_NETWORK socket785   80
driver25a
SOCKET_NETWORK socket786   80
driver25b
SOCKET_NETWORK socket787   80
driver25c
SOCKET_NETWORK socket788   80
driver25d
SOCKET_NETWORK socket789   80
driver25a
SOCKET_NETWORK socket790   80
driver25b
SOCKET_NETWORK socket791   80
driver25c
SOCKET_NETWORK socket792   80
driver25d
SOCKET_NETWORK socket793   80
driver25a
SOCKET_NETWORK socket794   80
driver25b
SOCKET_NETWORK socket795   80
driver25c
SOCKET_NETWORK socket796   80
driver25d
SOCKET_NETWORK socket797   80
driver25a
SOCKET_NETWORK socket798   80
driver25b
SOCKET_NETWORK socket799   80
driver25c
SOCKET_NETWORK socket800   80
driver25d
SOCKET_NETWORK socket801   80
driver26a
```

```
SOCKET_NETWORK socket802  80
driver26b
SOCKET_NETWORK socket803  80
driver26c
SOCKET_NETWORK socket804  80
driver26d
SOCKET_NETWORK socket805  80
driver26a
SOCKET_NETWORK socket806  80
driver26b
SOCKET_NETWORK socket807  80
driver26c
SOCKET_NETWORK socket808  80
driver26d
SOCKET_NETWORK socket809  80
driver26a
SOCKET_NETWORK socket810  80
driver26b
SOCKET_NETWORK socket811  80
driver26c
SOCKET_NETWORK socket812  80
driver26d
SOCKET_NETWORK socket813  80
driver26a
SOCKET_NETWORK socket814  80
driver26b
SOCKET_NETWORK socket815  80
driver26c
SOCKET_NETWORK socket816  80
driver26d
SOCKET_NETWORK socket817  80
driver26a
SOCKET_NETWORK socket818  80
driver26b
SOCKET_NETWORK socket819  80
driver26c
SOCKET_NETWORK socket820  80
driver26d
SOCKET_NETWORK socket821  80
driver26a
SOCKET_NETWORK socket822  80
driver26b
SOCKET_NETWORK socket823  80
driver26c
SOCKET_NETWORK socket824  80
driver26d
SOCKET_NETWORK socket825  80
driver26a
SOCKET_NETWORK socket826  80
driver26b
SOCKET_NETWORK socket827  80
driver26c
SOCKET_NETWORK socket828  80
driver26d
SOCKET_NETWORK socket829  80
driver26a
SOCKET_NETWORK socket830  80
driver26b
SOCKET_NETWORK socket831  80
driver26c
SOCKET_NETWORK socket832  80
driver26d
SOCKET_NETWORK socket833  80
driver27a
SOCKET_NETWORK socket834  80
driver27b
SOCKET_NETWORK socket835  80
driver27c
SOCKET_NETWORK socket836  80
driver27d
SOCKET_NETWORK socket837  80
driver27a
SOCKET_NETWORK socket838  80
driver27b
SOCKET_NETWORK socket839  80
driver27c
SOCKET_NETWORK socket840  80
driver27d
SOCKET_NETWORK socket841  80
driver27a
SOCKET_NETWORK socket842  80
driver27b
SOCKET_NETWORK socket843  80
driver27c
SOCKET_NETWORK socket844  80
driver27d
SOCKET_NETWORK socket845  80
driver27a
SOCKET_NETWORK socket846  80
driver27b
SOCKET_NETWORK socket847  80
driver27c
SOCKET_NETWORK socket848  80
driver27d
SOCKET_NETWORK socket849  80
driver27a
SOCKET_NETWORK socket850  80
driver27b
SOCKET_NETWORK socket851  80
driver27c
SOCKET_NETWORK socket852  80
driver27d
SOCKET_NETWORK socket853  80
driver27a
SOCKET_NETWORK socket854  80
driver27b
SOCKET_NETWORK socket855  80
driver27c
SOCKET_NETWORK socket856  80
driver27d
SOCKET_NETWORK socket857  80
driver27a
SOCKET_NETWORK socket858  80
driver27b
SOCKET_NETWORK socket859  80
driver27c
SOCKET_NETWORK socket860  80
driver27d
SOCKET_NETWORK socket861  80
driver27a
SOCKET_NETWORK socket862  80
driver27b
SOCKET_NETWORK socket863  80
driver27c
SOCKET_NETWORK socket864  80
driver27d
SOCKET_NETWORK socket865  80
driver28a
SOCKET_NETWORK socket866  80
driver28b
SOCKET_NETWORK socket867  80
driver28c
SOCKET_NETWORK socket868  80
driver28d
SOCKET_NETWORK socket869  80
driver28a
SOCKET_NETWORK socket870  80
driver28b
SOCKET_NETWORK socket871  80
driver28c
SOCKET_NETWORK socket872  80
driver28d
SOCKET_NETWORK socket873  80
driver28a
SOCKET_NETWORK socket874  80
driver28b
SOCKET_NETWORK socket875  80
driver28c
SOCKET_NETWORK socket876  80
driver28d
SOCKET_NETWORK socket877  80
driver28a
SOCKET_NETWORK socket878  80
driver28b
SOCKET_NETWORK socket879  80
driver28c
SOCKET_NETWORK socket880  80
driver28d
SOCKET_NETWORK socket881  80
driver28a
SOCKET_NETWORK socket882  80
driver28b
SOCKET_NETWORK socket883  80
driver28c
SOCKET_NETWORK socket884  80
driver28d
SOCKET_NETWORK socket885  80
driver28a
SOCKET_NETWORK socket886  80
driver28b
SOCKET_NETWORK socket887  80
driver28c
SOCKET_NETWORK socket888  80
driver28d
SOCKET_NETWORK socket889  80
driver28a
SOCKET_NETWORK socket890  80
driver28b
SOCKET_NETWORK socket891  80
driver28c
SOCKET_NETWORK socket892  80
driver28d
SOCKET_NETWORK socket893  80
driver28a
SOCKET_NETWORK socket894  80
driver28b
SOCKET_NETWORK socket895  80
driver28c
SOCKET_NETWORK socket896  80
driver28d
#elif MASTER_NUM8
SOCKET_NETWORK socket897  80
driver29a
SOCKET_NETWORK socket898  80
driver29b
SOCKET_NETWORK socket899  80
driver29c
SOCKET_NETWORK socket900  80
driver29d
SOCKET_NETWORK socket901  80
driver29a
SOCKET_NETWORK socket902  80
driver29b
SOCKET_NETWORK socket903  80
driver29c
SOCKET_NETWORK socket904  80
driver29d
SOCKET_NETWORK socket905  80
driver29a
SOCKET_NETWORK socket906  80
driver29b
SOCKET_NETWORK socket907  80
driver29c
SOCKET_NETWORK socket908  80
driver29d
SOCKET_NETWORK socket909  80
driver29a
SOCKET_NETWORK socket910  80
driver29b
SOCKET_NETWORK socket911  80
driver29c
SOCKET_NETWORK socket912  80
driver29d
SOCKET_NETWORK socket913  80
driver29a
SOCKET_NETWORK socket914  80
driver29b
SOCKET_NETWORK socket915  80
driver29c
SOCKET_NETWORK socket916  80
driver29d
SOCKET_NETWORK socket917  80
driver29a
SOCKET_NETWORK socket918  80
driver29b
SOCKET_NETWORK socket919  80
driver29c
SOCKET_NETWORK socket920  80
driver29d
SOCKET_NETWORK socket921  80
driver29a
SOCKET_NETWORK socket922  80
driver29b
SOCKET_NETWORK socket923  80
driver29c
SOCKET_NETWORK socket924  80
driver29d
```

```
SOCKET_NETWORK socket925   80
driver29a
SOCKET_NETWORK socket926   80
driver29b
SOCKET_NETWORK socket927   80
driver29c
SOCKET_NETWORK socket928   80
driver29d
SOCKET_NETWORK socket929   80
driver30a
SOCKET_NETWORK socket930   80
driver30b
SOCKET_NETWORK socket931   80
driver30c
SOCKET_NETWORK socket932   80
driver30d
SOCKET_NETWORK socket933   80
driver30a
SOCKET_NETWORK socket934   80
driver30b
SOCKET_NETWORK socket935   80
driver30c
SOCKET_NETWORK socket936   80
driver30d
SOCKET_NETWORK socket937   80
driver30a
SOCKET_NETWORK socket938   80
driver30b
SOCKET_NETWORK socket939   80
driver30c
SOCKET_NETWORK socket940   80
driver30d
SOCKET_NETWORK socket941   80
driver30a
SOCKET_NETWORK socket942   80
driver30b
SOCKET_NETWORK socket943   80
driver30c
SOCKET_NETWORK socket944   80
driver30d
SOCKET_NETWORK socket945   80
driver30a
SOCKET_NETWORK socket946   80
driver30b
SOCKET_NETWORK socket947   80
driver30c
SOCKET_NETWORK socket948   80
driver30d
SOCKET_NETWORK socket949   80
driver30a
SOCKET_NETWORK socket950   80
driver30b
SOCKET_NETWORK socket951   80
driver30c
SOCKET_NETWORK socket952   80
driver30d
SOCKET_NETWORK socket953   80
driver30a
SOCKET_NETWORK socket954   80
driver30b
SOCKET_NETWORK socket955   80
driver30c
SOCKET_NETWORK socket956   80
driver30d
SOCKET_NETWORK socket957   80
driver30a
SOCKET_NETWORK socket958   80
driver30b
SOCKET_NETWORK socket959   80
driver30c
SOCKET_NETWORK socket960   80
driver30d
SOCKET_NETWORK socket961   80
driver31a
SOCKET_NETWORK socket962   80
driver31b
SOCKET_NETWORK socket963   80
driver31c
SOCKET_NETWORK socket964   80
driver31d
SOCKET_NETWORK socket965   80
driver31a
SOCKET_NETWORK socket966   80
driver31b
SOCKET_NETWORK socket967   80
driver31c
SOCKET_NETWORK socket968   80
driver31d
SOCKET_NETWORK socket969   80
driver31a
SOCKET_NETWORK socket970   80
driver31b
SOCKET_NETWORK socket971   80
driver31c
SOCKET_NETWORK socket972   80
driver31d
SOCKET_NETWORK socket973   80
driver31a
SOCKET_NETWORK socket974   80
driver31b
SOCKET_NETWORK socket975   80
driver31c
SOCKET_NETWORK socket976   80
driver31d
SOCKET_NETWORK socket977   80
driver31a
SOCKET_NETWORK socket978   80
driver31b
SOCKET_NETWORK socket979   80
driver31c
SOCKET_NETWORK socket980   80
driver31d
SOCKET_NETWORK socket981   80
driver31a
SOCKET_NETWORK socket982   80
driver31b
SOCKET_NETWORK socket983   80
driver31c
SOCKET_NETWORK socket984   80
driver31d
SOCKET_NETWORK socket985   80
driver31a
SOCKET_NETWORK socket986   80
driver31b
SOCKET_NETWORK socket987   80
driver31c
SOCKET_NETWORK socket988   80
driver31d
SOCKET_NETWORK socket989   80
driver31a
SOCKET_NETWORK socket990   80
driver31b
SOCKET_NETWORK socket991   80
driver31c
SOCKET_NETWORK socket992   80
driver31d
SOCKET_NETWORK socket993   80
driver32a
SOCKET_NETWORK socket994   80
driver32b
SOCKET_NETWORK socket995   80
driver32c
SOCKET_NETWORK socket996   80
driver32d
SOCKET_NETWORK socket997   80
driver32a
SOCKET_NETWORK socket998   80
driver32b
SOCKET_NETWORK socket999   80
driver32c
SOCKET_NETWORK socket1000   80
driver32d
SOCKET_NETWORK socket1001   80
driver32a
SOCKET_NETWORK socket1002   80
driver32b
SOCKET_NETWORK socket1003   80
driver32c
SOCKET_NETWORK socket1004   80
driver32d
SOCKET_NETWORK socket1005   80
driver32a
SOCKET_NETWORK socket1006   80
driver32b
SOCKET_NETWORK socket1007   80
driver32c
SOCKET_NETWORK socket1008   80
driver32d
SOCKET_NETWORK socket1009   80
driver32a
SOCKET_NETWORK socket1010   80
driver32b
SOCKET_NETWORK socket1011   80
driver32c
SOCKET_NETWORK socket1012   80
driver32d
SOCKET_NETWORK socket1013   80
driver32a
SOCKET_NETWORK socket1014   80
driver32b
SOCKET_NETWORK socket1015   80
driver32c
SOCKET_NETWORK socket1016   80
driver32d
SOCKET_NETWORK socket1017   80
driver32a
SOCKET_NETWORK socket1018   80
driver32b
SOCKET_NETWORK socket1019   80
driver32c
SOCKET_NETWORK socket1020   80
driver32d
SOCKET_NETWORK socket1021   80
driver32a
SOCKET_NETWORK socket1022   80
driver32b
SOCKET_NETWORK socket1023   80
driver32c
SOCKET_NETWORK socket1024   80
driver32d
#elif MASTER_NUM9
SOCKET_NETWORK socket1025   80
driver33a
SOCKET_NETWORK socket1026   80
driver33b
SOCKET_NETWORK socket1027   80
driver33c
SOCKET_NETWORK socket1028   80
driver33d
SOCKET_NETWORK socket1029   80
driver33a
SOCKET_NETWORK socket1030   80
driver33b
SOCKET_NETWORK socket1031   80
driver33c
SOCKET_NETWORK socket1032   80
driver33d
SOCKET_NETWORK socket1033   80
driver33a
SOCKET_NETWORK socket1034   80
driver33b
SOCKET_NETWORK socket1035   80
driver33c
SOCKET_NETWORK socket1036   80
driver33d
SOCKET_NETWORK socket1037   80
driver33a
SOCKET_NETWORK socket1038   80
driver33b
SOCKET_NETWORK socket1039   80
driver33c
SOCKET_NETWORK socket1040   80
driver33d
SOCKET_NETWORK socket1041   80
driver33a
SOCKET_NETWORK socket1042   80
driver33b
SOCKET_NETWORK socket1043   80
driver33c
SOCKET_NETWORK socket1044   80
driver33d
SOCKET_NETWORK socket1045   80
driver33a
SOCKET_NETWORK socket1046   80
driver33b
SOCKET_NETWORK socket1047   80
driver33c
```

```
SOCKET_NETWORK socket1048 80
driver33d
SOCKET_NETWORK socket1049 80
driver33a
SOCKET_NETWORK socket1050 80
driver33b
SOCKET_NETWORK socket1051 80
driver33c
SOCKET_NETWORK socket1052 80
driver33d
SOCKET_NETWORK socket1053 80
driver33a
SOCKET_NETWORK socket1054 80
driver33b
SOCKET_NETWORK socket1055 80
driver33c
SOCKET_NETWORK socket1056 80
driver33d
SOCKET_NETWORK socket1057 80
driver34a
SOCKET_NETWORK socket1058 80
driver34b
SOCKET_NETWORK socket1059 80
driver34c
SOCKET_NETWORK socket1060 80
driver34d
SOCKET_NETWORK socket1061 80
driver34a
SOCKET_NETWORK socket1062 80
driver34b
SOCKET_NETWORK socket1063 80
driver34c
SOCKET_NETWORK socket1064 80
driver34d
SOCKET_NETWORK socket1065 80
driver34a
SOCKET_NETWORK socket1066 80
driver34b
SOCKET_NETWORK socket1067 80
driver34c
SOCKET_NETWORK socket1068 80
driver34d
SOCKET_NETWORK socket1069 80
driver34a
SOCKET_NETWORK socket1070 80
driver34b
SOCKET_NETWORK socket1071 80
driver34c
SOCKET_NETWORK socket1072 80
driver34d
SOCKET_NETWORK socket1073 80
driver34a
SOCKET_NETWORK socket1074 80
driver34b
SOCKET_NETWORK socket1075 80
driver34c
SOCKET_NETWORK socket1076 80
driver34d
SOCKET_NETWORK socket1077 80
driver34a
SOCKET_NETWORK socket1078 80
driver34b
SOCKET_NETWORK socket1079 80
driver34c
SOCKET_NETWORK socket1080 80
driver34d
SOCKET_NETWORK socket1081 80
driver34a
SOCKET_NETWORK socket1082 80
driver34b
SOCKET_NETWORK socket1083 80
driver34c
SOCKET_NETWORK socket1084 80
driver34d
SOCKET_NETWORK socket1085 80
driver34a
SOCKET_NETWORK socket1086 80
driver34b
SOCKET_NETWORK socket1087 80
driver34c
SOCKET_NETWORK socket1088 80
driver34d
SOCKET_NETWORK socket1089 80
driver35a
SOCKET_NETWORK socket1090 80
driver35b
SOCKET_NETWORK socket1091 80
driver35c
SOCKET_NETWORK socket1092 80
driver35d
SOCKET_NETWORK socket1093 80
driver35a
SOCKET_NETWORK socket1094 80
driver35b
SOCKET_NETWORK socket1095 80
driver35c
SOCKET_NETWORK socket1096 80
driver35d
SOCKET_NETWORK socket1097 80
driver35a
SOCKET_NETWORK socket1098 80
driver35b
SOCKET_NETWORK socket1099 80
driver35c
SOCKET_NETWORK socket1100 80
driver35d
SOCKET_NETWORK socket1101 80
driver35a
SOCKET_NETWORK socket1102 80
driver35b
SOCKET_NETWORK socket1103 80
driver35c
SOCKET_NETWORK socket1104 80
driver35d
SOCKET_NETWORK socket1105 80
driver35a
SOCKET_NETWORK socket1106 80
driver35b
SOCKET_NETWORK socket1107 80
driver35c
SOCKET_NETWORK socket1108 80
driver35d
SOCKET_NETWORK socket1109 80
driver35a
SOCKET_NETWORK socket1110 80
driver35b
SOCKET_NETWORK socket1111 80
driver35c
SOCKET_NETWORK socket1112 80
driver35d
SOCKET_NETWORK socket1113 80
driver35a
SOCKET_NETWORK socket1114 80
driver35b
SOCKET_NETWORK socket1115 80
driver35c
SOCKET_NETWORK socket1116 80
driver35d
SOCKET_NETWORK socket1117 80
driver35a
SOCKET_NETWORK socket1118 80
driver35b
SOCKET_NETWORK socket1119 80
driver35c
SOCKET_NETWORK socket1120 80
driver35d
SOCKET_NETWORK socket1121 80
driver36a
SOCKET_NETWORK socket1122 80
driver36b
SOCKET_NETWORK socket1123 80
driver36c
SOCKET_NETWORK socket1124 80
driver36d
SOCKET_NETWORK socket1125 80
driver36a
SOCKET_NETWORK socket1126 80
driver36b
SOCKET_NETWORK socket1127 80
driver36c
SOCKET_NETWORK socket1128 80
driver36d
SOCKET_NETWORK socket1129 80
driver36a
SOCKET_NETWORK socket1130 80
driver36b
SOCKET_NETWORK socket1131 80
driver36c
SOCKET_NETWORK socket1132 80
driver36d
SOCKET_NETWORK socket1133 80
driver36a
SOCKET_NETWORK socket1134 80
driver36b
SOCKET_NETWORK socket1135 80
driver36c
SOCKET_NETWORK socket1136 80
driver36d
SOCKET_NETWORK socket1137 80
driver36a
SOCKET_NETWORK socket1138 80
driver36b
SOCKET_NETWORK socket1139 80
driver36c
SOCKET_NETWORK socket1140 80
driver36d
SOCKET_NETWORK socket1141 80
driver36a
SOCKET_NETWORK socket1142 80
driver36b
SOCKET_NETWORK socket1143 80
driver36c
SOCKET_NETWORK socket1144 80
driver36d
SOCKET_NETWORK socket1145 80
driver36a
SOCKET_NETWORK socket1146 80
driver36b
SOCKET_NETWORK socket1147 80
driver36c
SOCKET_NETWORK socket1148 80
driver36d
SOCKET_NETWORK socket1149 80
driver36a
SOCKET_NETWORK socket1150 80
driver36b
SOCKET_NETWORK socket1151 80
driver36c
SOCKET_NETWORK socket1152 80
driver36d
#elif MASTER_NUM10
SOCKET_NETWORK socket1153 80
driver37a
SOCKET_NETWORK socket1154 80
driver37b
SOCKET_NETWORK socket1155 80
driver37c
SOCKET_NETWORK socket1156 80
driver37d
SOCKET_NETWORK socket1157 80
driver37a
SOCKET_NETWORK socket1158 80
driver37b
SOCKET_NETWORK socket1159 80
driver37c
SOCKET_NETWORK socket1160 80
driver37d
SOCKET_NETWORK socket1161 80
driver37a
SOCKET_NETWORK socket1162 80
driver37b
SOCKET_NETWORK socket1163 80
driver37c
SOCKET_NETWORK socket1164 80
driver37d
SOCKET_NETWORK socket1165 80
driver37a
SOCKET_NETWORK socket1166 80
driver37b
SOCKET_NETWORK socket1167 80
driver37c
SOCKET_NETWORK socket1168 80
driver37d
SOCKET_NETWORK socket1169 80
driver37a
SOCKET_NETWORK socket1170 80
driver37b
```

```
SOCKET_NETWORK socket1171 80
driver37c
SOCKET_NETWORK socket1172 80
driver37d
SOCKET_NETWORK socket1173 80
driver37a
SOCKET_NETWORK socket1174 80
driver37b
SOCKET_NETWORK socket1175 80
driver37c
SOCKET_NETWORK socket1176 80
driver37d
SOCKET_NETWORK socket1177 80
driver37a
SOCKET_NETWORK socket1178 80
driver37b
SOCKET_NETWORK socket1179 80
driver37c
SOCKET_NETWORK socket1180 80
driver37d
SOCKET_NETWORK socket1181 80
driver37a
SOCKET_NETWORK socket1182 80
driver37b
SOCKET_NETWORK socket1183 80
driver37c
SOCKET_NETWORK socket1184 80
driver37d
SOCKET_NETWORK socket1185 80
driver38a
SOCKET_NETWORK socket1186 80
driver38b
SOCKET_NETWORK socket1187 80
driver38c
SOCKET_NETWORK socket1188 80
driver38d
SOCKET_NETWORK socket1189 80
driver38a
SOCKET_NETWORK socket1190 80
driver38b
SOCKET_NETWORK socket1191 80
driver38c
SOCKET_NETWORK socket1192 80
driver38d
SOCKET_NETWORK socket1193 80
driver38a
SOCKET_NETWORK socket1194 80
driver38b
SOCKET_NETWORK socket1195 80
driver38c
SOCKET_NETWORK socket1196 80
driver38d
SOCKET_NETWORK socket1197 80
driver38a
SOCKET_NETWORK socket1198 80
driver38b
SOCKET_NETWORK socket1199 80
driver38c
SOCKET_NETWORK socket1200 80
driver38d
SOCKET_NETWORK socket1201 80
driver38a
SOCKET_NETWORK socket1202 80
driver38b
SOCKET_NETWORK socket1203 80
driver38c
SOCKET_NETWORK socket1204 80
driver38d
SOCKET_NETWORK socket1205 80
driver38a
SOCKET_NETWORK socket1206 80
driver38b
SOCKET_NETWORK socket1207 80
driver38c
SOCKET_NETWORK socket1208 80
driver38d
SOCKET_NETWORK socket1209 80
driver38a
SOCKET_NETWORK socket1210 80
driver38b
SOCKET_NETWORK socket1211 80
driver38c
SOCKET_NETWORK socket1212 80
driver38d
SOCKET_NETWORK socket1213 80
driver38a
SOCKET_NETWORK socket1214 80
driver38b
SOCKET_NETWORK socket1215 80
driver38c
SOCKET_NETWORK socket1216 80
driver38d
SOCKET_NETWORK socket1217 80
driver39a
SOCKET_NETWORK socket1218 80
driver39b
SOCKET_NETWORK socket1219 80
driver39c
SOCKET_NETWORK socket1220 80
driver39d
SOCKET_NETWORK socket1221 80
driver39a
SOCKET_NETWORK socket1222 80
driver39b
SOCKET_NETWORK socket1223 80
driver39c
SOCKET_NETWORK socket1224 80
driver39d
SOCKET_NETWORK socket1225 80
driver39a
SOCKET_NETWORK socket1226 80
driver39b
SOCKET_NETWORK socket1227 80
driver39c
SOCKET_NETWORK socket1228 80
driver39d
SOCKET_NETWORK socket1229 80
driver39a
SOCKET_NETWORK socket1230 80
driver39b
SOCKET_NETWORK socket1231 80
driver39c
SOCKET_NETWORK socket1232 80
driver39d
SOCKET_NETWORK socket1233 80
driver39a
SOCKET_NETWORK socket1234 80
driver39b
SOCKET_NETWORK socket1235 80
driver39c
SOCKET_NETWORK socket1236 80
driver39d
SOCKET_NETWORK socket1237 80
driver39a
SOCKET_NETWORK socket1238 80
driver39b
SOCKET_NETWORK socket1239 80
driver39c
SOCKET_NETWORK socket1240 80
driver39d
SOCKET_NETWORK socket1241 80
driver39a
SOCKET_NETWORK socket1242 80
driver39b
SOCKET_NETWORK socket1243 80
driver39c
SOCKET_NETWORK socket1244 80
driver39d
SOCKET_NETWORK socket1245 80
driver39a
SOCKET_NETWORK socket1246 80
driver39b
SOCKET_NETWORK socket1247 80
driver39c
SOCKET_NETWORK socket1248 80
driver39d
SOCKET_NETWORK socket1249 80
driver40a
SOCKET_NETWORK socket1250 80
driver40b
SOCKET_NETWORK socket1251 80
driver40c
SOCKET_NETWORK socket1252 80
driver40d
SOCKET_NETWORK socket1253 80
driver40a
SOCKET_NETWORK socket1254 80
driver40b
SOCKET_NETWORK socket1255 80
driver40c
SOCKET_NETWORK socket1256 80
driver40d
SOCKET_NETWORK socket1257 80
driver40a
SOCKET_NETWORK socket1258 80
driver40b
SOCKET_NETWORK socket1259 80
driver40c
SOCKET_NETWORK socket1260 80
driver40d
SOCKET_NETWORK socket1261 80
driver40a
SOCKET_NETWORK socket1262 80
driver40b
SOCKET_NETWORK socket1263 80
driver40c
SOCKET_NETWORK socket1264 80
driver40d
SOCKET_NETWORK socket1265 80
driver40a
SOCKET_NETWORK socket1266 80
driver40b
SOCKET_NETWORK socket1267 80
driver40c
SOCKET_NETWORK socket1268 80
driver40d
SOCKET_NETWORK socket1269 80
driver40a
SOCKET_NETWORK socket1270 80
driver40b
SOCKET_NETWORK socket1271 80
driver40c
SOCKET_NETWORK socket1272 80
driver40d
SOCKET_NETWORK socket1273 80
driver40a
SOCKET_NETWORK socket1274 80
driver40b
SOCKET_NETWORK socket1275 80
driver40c
SOCKET_NETWORK socket1276 80
driver40d
SOCKET_NETWORK socket1277 80
driver40a
SOCKET_NETWORK socket1278 80
driver40b
SOCKET_NETWORK socket1279 80
driver40c
SOCKET_NETWORK socket1280 80
driver40d
#elif MASTER_NUM11
SOCKET_NETWORK socket1281 80
driver41a
SOCKET_NETWORK socket1282 80
driver41b
SOCKET_NETWORK socket1283 80
driver41c
SOCKET_NETWORK socket1284 80
driver41d
SOCKET_NETWORK socket1285 80
driver41a
SOCKET_NETWORK socket1286 80
driver41b
SOCKET_NETWORK socket1287 80
driver41c
SOCKET_NETWORK socket1288 80
driver41d
SOCKET_NETWORK socket1289 80
driver41a
SOCKET_NETWORK socket1290 80
driver41b
SOCKET_NETWORK socket1291 80
driver41c
SOCKET_NETWORK socket1292 80
driver41d
SOCKET_NETWORK socket1293 80
driver41a
```

```
SOCKET_NETWORK socket1294 80
driver41b
SOCKET_NETWORK socket1295 80
driver41c
SOCKET_NETWORK socket1296 80
driver41d
SOCKET_NETWORK socket1297 80
driver41a
SOCKET_NETWORK socket1298 80
driver41b
SOCKET_NETWORK socket1299 80
driver41c
SOCKET_NETWORK socket1300 80
driver41d
SOCKET_NETWORK socket1301 80
driver41a
SOCKET_NETWORK socket1302 80
driver41b
SOCKET_NETWORK socket1303 80
driver41c
SOCKET_NETWORK socket1304 80
driver41d
SOCKET_NETWORK socket1305 80
driver41a
SOCKET_NETWORK socket1306 80
driver41b
SOCKET_NETWORK socket1307 80
driver41c
SOCKET_NETWORK socket1308 80
driver41d
SOCKET_NETWORK socket1309 80
driver41a
SOCKET_NETWORK socket1310 80
driver41b
SOCKET_NETWORK socket1311 80
driver41c
SOCKET_NETWORK socket1312 80
driver41d
SOCKET_NETWORK socket1313 80
driver42a
SOCKET_NETWORK socket1314 80
driver42b
SOCKET_NETWORK socket1315 80
driver42c
SOCKET_NETWORK socket1316 80
driver42d
SOCKET_NETWORK socket1317 80
driver42a
SOCKET_NETWORK socket1318 80
driver42b
SOCKET_NETWORK socket1319 80
driver42c
SOCKET_NETWORK socket1320 80
driver42d
SOCKET_NETWORK socket1321 80
driver42a
SOCKET_NETWORK socket1322 80
driver42b
SOCKET_NETWORK socket1323 80
driver42c
SOCKET_NETWORK socket1324 80
driver42d
SOCKET_NETWORK socket1325 80
driver42a
SOCKET_NETWORK socket1326 80
driver42b
SOCKET_NETWORK socket1327 80
driver42c
SOCKET_NETWORK socket1328 80
driver42d
SOCKET_NETWORK socket1329 80
driver42a
SOCKET_NETWORK socket1330 80
driver42b
SOCKET_NETWORK socket1331 80
driver42c
SOCKET_NETWORK socket1332 80
driver42d
SOCKET_NETWORK socket1333 80
driver42a
SOCKET_NETWORK socket1334 80
driver42b
SOCKET_NETWORK socket1335 80
driver42c
SOCKET_NETWORK socket1336 80
driver42d
SOCKET_NETWORK socket1337 80
driver42a
SOCKET_NETWORK socket1338 80
driver42b
SOCKET_NETWORK socket1339 80
driver42c
SOCKET_NETWORK socket1340 80
driver42d
SOCKET_NETWORK socket1341 80
driver42a
SOCKET_NETWORK socket1342 80
driver42b
SOCKET_NETWORK socket1343 80
driver42c
SOCKET_NETWORK socket1344 80
driver42d
SOCKET_NETWORK socket1345 80
driver43a
SOCKET_NETWORK socket1346 80
driver43b
SOCKET_NETWORK socket1347 80
driver43c
SOCKET_NETWORK socket1348 80
driver43d
SOCKET_NETWORK socket1349 80
driver43a
SOCKET_NETWORK socket1350 80
driver43b
SOCKET_NETWORK socket1351 80
driver43c
SOCKET_NETWORK socket1352 80
driver43d
SOCKET_NETWORK socket1353 80
driver43a
SOCKET_NETWORK socket1354 80
driver43b
SOCKET_NETWORK socket1355 80
driver43c
SOCKET_NETWORK socket1356 80
driver43d
SOCKET_NETWORK socket1357 80
driver43a
SOCKET_NETWORK socket1358 80
driver43b
SOCKET_NETWORK socket1359 80
driver43c
SOCKET_NETWORK socket1360 80
driver43d
SOCKET_NETWORK socket1361 80
driver43a
SOCKET_NETWORK socket1362 80
driver43b
SOCKET_NETWORK socket1363 80
driver43c
SOCKET_NETWORK socket1364 80
driver43d
SOCKET_NETWORK socket1365 80
driver43a
SOCKET_NETWORK socket1366 80
driver43b
SOCKET_NETWORK socket1367 80
driver43c
SOCKET_NETWORK socket1368 80
driver43d
SOCKET_NETWORK socket1369 80
driver43a
SOCKET_NETWORK socket1370 80
driver43b
SOCKET_NETWORK socket1371 80
driver43c
SOCKET_NETWORK socket1372 80
driver43d
SOCKET_NETWORK socket1373 80
driver43a
SOCKET_NETWORK socket1374 80
driver43b
SOCKET_NETWORK socket1375 80
driver43c
SOCKET_NETWORK socket1376 80
driver43d
SOCKET_NETWORK socket1377 80
driver44a
SOCKET_NETWORK socket1378 80
driver44b
SOCKET_NETWORK socket1379 80
driver44c
SOCKET_NETWORK socket1380 80
driver44d
SOCKET_NETWORK socket1381 80
driver44a
SOCKET_NETWORK socket1382 80
driver44b
SOCKET_NETWORK socket1383 80
driver44c
SOCKET_NETWORK socket1384 80
driver44d
SOCKET_NETWORK socket1385 80
driver44a
SOCKET_NETWORK socket1386 80
driver44b
SOCKET_NETWORK socket1387 80
driver44c
SOCKET_NETWORK socket1388 80
driver44d
SOCKET_NETWORK socket1389 80
driver44a
SOCKET_NETWORK socket1390 80
driver44b
SOCKET_NETWORK socket1391 80
driver44c
SOCKET_NETWORK socket1392 80
driver44d
SOCKET_NETWORK socket1393 80
driver44a
SOCKET_NETWORK socket1394 80
driver44b
SOCKET_NETWORK socket1395 80
driver44c
SOCKET_NETWORK socket1396 80
driver44d
SOCKET_NETWORK socket1397 80
driver44a
SOCKET_NETWORK socket1398 80
driver44b
SOCKET_NETWORK socket1399 80
driver44c
SOCKET_NETWORK socket1400 80
driver44d
SOCKET_NETWORK socket1401 80
driver44a
SOCKET_NETWORK socket1402 80
driver44b
SOCKET_NETWORK socket1403 80
driver44c
SOCKET_NETWORK socket1404 80
driver44d
SOCKET_NETWORK socket1405 80
driver44a
SOCKET_NETWORK socket1406 80
driver44b
SOCKET_NETWORK socket1407 80
driver44c
SOCKET_NETWORK socket1408 80
driver44d
#elif MASTER_NUM12
SOCKET_NETWORK socket1409 80
driver45a
SOCKET_NETWORK socket1410 80
driver45b
SOCKET_NETWORK socket1411 80
driver45c
SOCKET_NETWORK socket1412 80
driver45d
SOCKET_NETWORK socket1413 80
driver45a
SOCKET_NETWORK socket1414 80
driver45b
SOCKET_NETWORK socket1415 80
driver45c
SOCKET_NETWORK socket1416 80
driver45d
```

```
SOCKET_NETWORK socket1417 80
driver45a
SOCKET_NETWORK socket1418 80
driver45b
SOCKET_NETWORK socket1419 80
driver45c
SOCKET_NETWORK socket1420 80
driver45d
SOCKET_NETWORK socket1421 80
driver45a
SOCKET_NETWORK socket1422 80
driver45b
SOCKET_NETWORK socket1423 80
driver45c
SOCKET_NETWORK socket1424 80
driver45d
SOCKET_NETWORK socket1425 80
driver45a
SOCKET_NETWORK socket1426 80
driver45b
SOCKET_NETWORK socket1427 80
driver45c
SOCKET_NETWORK socket1428 80
driver45d
SOCKET_NETWORK socket1429 80
driver45a
SOCKET_NETWORK socket1430 80
driver45b
SOCKET_NETWORK socket1431 80
driver45c
SOCKET_NETWORK socket1432 80
driver45d
SOCKET_NETWORK socket1433 80
driver45a
SOCKET_NETWORK socket1434 80
driver45b
SOCKET_NETWORK socket1435 80
driver45c
SOCKET_NETWORK socket1436 80
driver45d
SOCKET_NETWORK socket1437 80
driver45a
SOCKET_NETWORK socket1438 80
driver45b
SOCKET_NETWORK socket1439 80
driver45c
SOCKET_NETWORK socket1440 80
driver45d
SOCKET_NETWORK socket1441 80
driver46a
SOCKET_NETWORK socket1442 80
driver46b
SOCKET_NETWORK socket1443 80
driver46c
SOCKET_NETWORK socket1444 80
driver46d
SOCKET_NETWORK socket1445 80
driver46a
SOCKET_NETWORK socket1446 80
driver46b
SOCKET_NETWORK socket1447 80
driver46c
SOCKET_NETWORK socket1448 80
driver46d
SOCKET_NETWORK socket1449 80
driver46a
SOCKET_NETWORK socket1450 80
driver46b
SOCKET_NETWORK socket1451 80
driver46c
SOCKET_NETWORK socket1452 80
driver46d
SOCKET_NETWORK socket1453 80
driver46a
SOCKET_NETWORK socket1454 80
driver46b
SOCKET_NETWORK socket1455 80
driver46c
SOCKET_NETWORK socket1456 80
driver46d
SOCKET_NETWORK socket1457 80
driver46a
SOCKET_NETWORK socket1458 80
driver46b
SOCKET_NETWORK socket1459 80
driver46c
SOCKET_NETWORK socket1460 80
driver46d
SOCKET_NETWORK socket1461 80
driver46a
SOCKET_NETWORK socket1462 80
driver46b
SOCKET_NETWORK socket1463 80
driver46c
SOCKET_NETWORK socket1464 80
driver46d
SOCKET_NETWORK socket1465 80
driver46a
SOCKET_NETWORK socket1466 80
driver46b
SOCKET_NETWORK socket1467 80
driver46c
SOCKET_NETWORK socket1468 80
driver46d
SOCKET_NETWORK socket1469 80
driver46a
SOCKET_NETWORK socket1470 80
driver46b
SOCKET_NETWORK socket1471 80
driver46c
SOCKET_NETWORK socket1472 80
driver46d
SOCKET_NETWORK socket1473 80
driver47a
SOCKET_NETWORK socket1474 80
driver47b
SOCKET_NETWORK socket1475 80
driver47c
SOCKET_NETWORK socket1476 80
driver47d
SOCKET_NETWORK socket1477 80
driver47a
SOCKET_NETWORK socket1478 80
driver47b
SOCKET_NETWORK socket1479 80
driver47c
SOCKET_NETWORK socket1480 80
driver47d
SOCKET_NETWORK socket1481 80
driver47a
SOCKET_NETWORK socket1482 80
driver47b
SOCKET_NETWORK socket1483 80
driver47c
SOCKET_NETWORK socket1484 80
driver47d
SOCKET_NETWORK socket1485 80
driver47a
SOCKET_NETWORK socket1486 80
driver47b
SOCKET_NETWORK socket1487 80
driver47c
SOCKET_NETWORK socket1488 80
driver47d
SOCKET_NETWORK socket1489 80
driver47a
SOCKET_NETWORK socket1490 80
driver47b
SOCKET_NETWORK socket1491 80
driver47c
SOCKET_NETWORK socket1492 80
driver47d
SOCKET_NETWORK socket1493 80
driver47a
SOCKET_NETWORK socket1494 80
driver47b
SOCKET_NETWORK socket1495 80
driver47c
SOCKET_NETWORK socket1496 80
driver47d
SOCKET_NETWORK socket1497 80
driver47a
SOCKET_NETWORK socket1498 80
driver47b
SOCKET_NETWORK socket1499 80
driver47c
SOCKET_NETWORK socket1500 80
driver47d
SOCKET_NETWORK socket1501 80
driver47a
SOCKET_NETWORK socket1502 80
driver47b
SOCKET_NETWORK socket1503 80
driver47c
SOCKET_NETWORK socket1504 80
driver47d
SOCKET_NETWORK socket1505 80
driver48a
SOCKET_NETWORK socket1506 80
driver48b
SOCKET_NETWORK socket1507 80
driver48c
SOCKET_NETWORK socket1508 80
driver48d
SOCKET_NETWORK socket1509 80
driver48a
SOCKET_NETWORK socket1510 80
driver48b
SOCKET_NETWORK socket1511 80
driver48c
SOCKET_NETWORK socket1512 80
driver48d
SOCKET_NETWORK socket1513 80
driver48a
SOCKET_NETWORK socket1514 80
driver48b
SOCKET_NETWORK socket1515 80
driver48c
SOCKET_NETWORK socket1516 80
driver48d
SOCKET_NETWORK socket1517 80
driver48a
SOCKET_NETWORK socket1518 80
driver48b
SOCKET_NETWORK socket1519 80
driver48c
SOCKET_NETWORK socket1520 80
driver48d
SOCKET_NETWORK socket1521 80
driver48a
SOCKET_NETWORK socket1522 80
driver48b
SOCKET_NETWORK socket1523 80
driver48c
SOCKET_NETWORK socket1524 80
driver48d
SOCKET_NETWORK socket1525 80
driver48a
SOCKET_NETWORK socket1526 80
driver48b
SOCKET_NETWORK socket1527 80
driver48c
SOCKET_NETWORK socket1528 80
driver48d
SOCKET_NETWORK socket1529 80
driver48a
SOCKET_NETWORK socket1530 80
driver48b
SOCKET_NETWORK socket1531 80
driver48c
SOCKET_NETWORK socket1532 80
driver48d
SOCKET_NETWORK socket1533 80
driver48a
SOCKET_NETWORK socket1534 80
driver48b
SOCKET_NETWORK socket1535 80
driver48c
SOCKET_NETWORK socket1536 80
driver48d
#elif MASTER_NUM13
SOCKET_NETWORK socket1537 80
driver49a
SOCKET_NETWORK socket1538 80
driver49b
SOCKET_NETWORK socket1539 80
driver49c
```

SOCKET_NETWORK socket1540 80
driver49d
SOCKET_NETWORK socket1541 80
driver49a
SOCKET_NETWORK socket1542 80
driver49b
SOCKET_NETWORK socket1543 80
driver49c
SOCKET_NETWORK socket1544 80
driver49d
SOCKET_NETWORK socket1545 80
driver49a
SOCKET_NETWORK socket1546 80
driver49b
SOCKET_NETWORK socket1547 80
driver49c
SOCKET_NETWORK socket1548 80
driver49d
SOCKET_NETWORK socket1549 80
driver49a
SOCKET_NETWORK socket1550 80
driver49b
SOCKET_NETWORK socket1551 80
driver49c
SOCKET_NETWORK socket1552 80
driver49d
SOCKET_NETWORK socket1553 80
driver49a
SOCKET_NETWORK socket1554 80
driver49b
SOCKET_NETWORK socket1555 80
driver49c
SOCKET_NETWORK socket1556 80
driver49d
SOCKET_NETWORK socket1557 80
driver49a
SOCKET_NETWORK socket1558 80
driver49b
SOCKET_NETWORK socket1559 80
driver49c
SOCKET_NETWORK socket1560 80
driver49d
SOCKET_NETWORK socket1561 80
driver49a
SOCKET_NETWORK socket1562 80
driver49b
SOCKET_NETWORK socket1563 80
driver49c
SOCKET_NETWORK socket1564 80
driver49d
SOCKET_NETWORK socket1565 80
driver49a
SOCKET_NETWORK socket1566 80
driver49b
SOCKET_NETWORK socket1567 80
driver49c
SOCKET_NETWORK socket1568 80
driver49d
SOCKET_NETWORK socket1569 80
driver50a
SOCKET_NETWORK socket1570 80
driver50b
SOCKET_NETWORK socket1571 80
driver50c
SOCKET_NETWORK socket1572 80
driver50d
SOCKET_NETWORK socket1573 80
driver50a
SOCKET_NETWORK socket1574 80
driver50b
SOCKET_NETWORK socket1575 80
driver50c
SOCKET_NETWORK socket1576 80
driver50d
SOCKET_NETWORK socket1577 80
driver50a
SOCKET_NETWORK socket1578 80
driver50b
SOCKET_NETWORK socket1579 80
driver50c
SOCKET_NETWORK socket1580 80
driver50d
SOCKET_NETWORK socket1581 80
driver50a
SOCKET_NETWORK socket1582 80
driver50b
SOCKET_NETWORK socket1583 80
driver50c
SOCKET_NETWORK socket1584 80
driver50d
SOCKET_NETWORK socket1585 80
driver50a
SOCKET_NETWORK socket1586 80
driver50b
SOCKET_NETWORK socket1587 80
driver50c
SOCKET_NETWORK socket1588 80
driver50d
SOCKET_NETWORK socket1589 80
driver50a
SOCKET_NETWORK socket1590 80
driver50b
SOCKET_NETWORK socket1591 80
driver50c
SOCKET_NETWORK socket1592 80
driver50d
SOCKET_NETWORK socket1593 80
driver50a
SOCKET_NETWORK socket1594 80
driver50b
SOCKET_NETWORK socket1595 80
driver50c
SOCKET_NETWORK socket1596 80
driver50d
SOCKET_NETWORK socket1597 80
driver50a
SOCKET_NETWORK socket1598 80
driver50b
SOCKET_NETWORK socket1599 80
driver50c
SOCKET_NETWORK socket1600 80
driver50d
SOCKET_NETWORK socket1601 80
driver51a
SOCKET_NETWORK socket1602 80
driver51b
SOCKET_NETWORK socket1603 80
driver51c
SOCKET_NETWORK socket1604 80
driver51d
SOCKET_NETWORK socket1605 80
driver51a
SOCKET_NETWORK socket1606 80
driver51b
SOCKET_NETWORK socket1607 80
driver51c
SOCKET_NETWORK socket1608 80
driver51d
SOCKET_NETWORK socket1609 80
driver51a
SOCKET_NETWORK socket1610 80
driver51b
SOCKET_NETWORK socket1611 80
driver51c
SOCKET_NETWORK socket1612 80
driver51d
SOCKET_NETWORK socket1613 80
driver51a
SOCKET_NETWORK socket1614 80
driver51b
SOCKET_NETWORK socket1615 80
driver51c
SOCKET_NETWORK socket1616 80
driver51d
SOCKET_NETWORK socket1617 80
driver51a
SOCKET_NETWORK socket1618 80
driver51b
SOCKET_NETWORK socket1619 80
driver51c
SOCKET_NETWORK socket1620 80
driver51d
SOCKET_NETWORK socket1621 80
driver51a
SOCKET_NETWORK socket1622 80
driver51b
SOCKET_NETWORK socket1623 80
driver51c
SOCKET_NETWORK socket1624 80
driver51d
SOCKET_NETWORK socket1625 80
driver51a
SOCKET_NETWORK socket1626 80
driver51b
SOCKET_NETWORK socket1627 80
driver51c
SOCKET_NETWORK socket1628 80
driver51d
SOCKET_NETWORK socket1629 80
driver51a
SOCKET_NETWORK socket1630 80
driver51b
SOCKET_NETWORK socket1631 80
driver51c
SOCKET_NETWORK socket1632 80
driver51d
SOCKET_NETWORK socket1633 80
driver52a
SOCKET_NETWORK socket1634 80
driver52b
SOCKET_NETWORK socket1635 80
driver52c
SOCKET_NETWORK socket1636 80
driver52d
SOCKET_NETWORK socket1637 80
driver52a
SOCKET_NETWORK socket1638 80
driver52b
SOCKET_NETWORK socket1639 80
driver52c
SOCKET_NETWORK socket1640 80
driver52d
SOCKET_NETWORK socket1641 80
driver52a
SOCKET_NETWORK socket1642 80
driver52b
SOCKET_NETWORK socket1643 80
driver52c
SOCKET_NETWORK socket1644 80
driver52d
SOCKET_NETWORK socket1645 80
driver52a
SOCKET_NETWORK socket1646 80
driver52b
SOCKET_NETWORK socket1647 80
driver52c
SOCKET_NETWORK socket1648 80
driver52d
SOCKET_NETWORK socket1649 80
driver52a
SOCKET_NETWORK socket1650 80
driver52b
SOCKET_NETWORK socket1651 80
driver52c
SOCKET_NETWORK socket1652 80
driver52d
SOCKET_NETWORK socket1653 80
driver52a
SOCKET_NETWORK socket1654 80
driver52b
SOCKET_NETWORK socket1655 80
driver52c
SOCKET_NETWORK socket1656 80
driver52d
SOCKET_NETWORK socket1657 80
driver52a
SOCKET_NETWORK socket1658 80
driver52b
SOCKET_NETWORK socket1659 80
driver52c
SOCKET_NETWORK socket1660 80
driver52d
SOCKET_NETWORK socket1661 80
driver52a
SOCKET_NETWORK socket1662 80
driver52b

```
SOCKET_NETWORK socket1663  80
driver52c
SOCKET_NETWORK socket1664  80
driver52d
#elif MASTER_NUM14
SOCKET_NETWORK socket1665  80
driver53a
SOCKET_NETWORK socket1666  80
driver53b
SOCKET_NETWORK socket1667  80
driver53c
SOCKET_NETWORK socket1668  80
driver53d
SOCKET_NETWORK socket1669  80
driver53a
SOCKET_NETWORK socket1670  80
driver53b
SOCKET_NETWORK socket1671  80
driver53c
SOCKET_NETWORK socket1672  80
driver53d
SOCKET_NETWORK socket1673  80
driver53a
SOCKET_NETWORK socket1674  80
driver53b
SOCKET_NETWORK socket1675  80
driver53c
SOCKET_NETWORK socket1676  80
driver53d
SOCKET_NETWORK socket1677  80
driver53a
SOCKET_NETWORK socket1678  80
driver53b
SOCKET_NETWORK socket1679  80
driver53c
SOCKET_NETWORK socket1680  80
driver53d
SOCKET_NETWORK socket1681  80
driver53a
SOCKET_NETWORK socket1682  80
driver53b
SOCKET_NETWORK socket1683  80
driver53c
SOCKET_NETWORK socket1684  80
driver53d
SOCKET_NETWORK socket1685  80
driver53a
SOCKET_NETWORK socket1686  80
driver53b
SOCKET_NETWORK socket1687  80
driver53c
SOCKET_NETWORK socket1688  80
driver53d
SOCKET_NETWORK socket1689  80
driver53a
SOCKET_NETWORK socket1690  80
driver53b
SOCKET_NETWORK socket1691  80
driver53c
SOCKET_NETWORK socket1692  80
driver53d
SOCKET_NETWORK socket1693  80
driver53a
SOCKET_NETWORK socket1694  80
driver53b
SOCKET_NETWORK socket1695  80
driver53c
SOCKET_NETWORK socket1696  80
driver53d
SOCKET_NETWORK socket1697  80
driver54a
SOCKET_NETWORK socket1698  80
driver54b
SOCKET_NETWORK socket1699  80
driver54c
SOCKET_NETWORK socket1700  80
driver54d
SOCKET_NETWORK socket1701  80
driver54a
SOCKET_NETWORK socket1702  80
driver54b
SOCKET_NETWORK socket1703  80
driver54c
SOCKET_NETWORK socket1704  80
driver54d
SOCKET_NETWORK socket1705  80
driver54a
SOCKET_NETWORK socket1706  80
driver54b
SOCKET_NETWORK socket1707  80
driver54c
SOCKET_NETWORK socket1708  80
driver54d
SOCKET_NETWORK socket1709  80
driver54a
SOCKET_NETWORK socket1710  80
driver54b
SOCKET_NETWORK socket1711  80
driver54c
SOCKET_NETWORK socket1712  80
driver54d
SOCKET_NETWORK socket1713  80
driver54a
SOCKET_NETWORK socket1714  80
driver54b
SOCKET_NETWORK socket1715  80
driver54c
SOCKET_NETWORK socket1716  80
driver54d
SOCKET_NETWORK socket1717  80
driver54a
SOCKET_NETWORK socket1718  80
driver54b
SOCKET_NETWORK socket1719  80
driver54c
SOCKET_NETWORK socket1720  80
driver54d
SOCKET_NETWORK socket1721  80
driver54a
SOCKET_NETWORK socket1722  80
driver54b
SOCKET_NETWORK socket1723  80
driver54c
SOCKET_NETWORK socket1724  80
driver54d
SOCKET_NETWORK socket1725  80
driver54a
SOCKET_NETWORK socket1726  80
driver54b
SOCKET_NETWORK socket1727  80
driver54c
SOCKET_NETWORK socket1728  80
driver54d
SOCKET_NETWORK socket1729  80
driver55a
SOCKET_NETWORK socket1730  80
driver55b
SOCKET_NETWORK socket1731  80
driver55c
SOCKET_NETWORK socket1732  80
driver55d
SOCKET_NETWORK socket1733  80
driver55a
SOCKET_NETWORK socket1734  80
driver55b
SOCKET_NETWORK socket1735  80
driver55c
SOCKET_NETWORK socket1736  80
driver55d
SOCKET_NETWORK socket1737  80
driver55a
SOCKET_NETWORK socket1738  80
driver55b
SOCKET_NETWORK socket1739  80
driver55c
SOCKET_NETWORK socket1740  80
driver55d
SOCKET_NETWORK socket1741  80
driver55a
SOCKET_NETWORK socket1742  80
driver55b
SOCKET_NETWORK socket1743  80
driver55c
SOCKET_NETWORK socket1744  80
driver55d
SOCKET_NETWORK socket1745  80
driver55a
SOCKET_NETWORK socket1746  80
driver55b
SOCKET_NETWORK socket1747  80
driver55c
SOCKET_NETWORK socket1748  80
driver55d
SOCKET_NETWORK socket1749  80
driver55a
SOCKET_NETWORK socket1750  80
driver55b
SOCKET_NETWORK socket1751  80
driver55c
SOCKET_NETWORK socket1752  80
driver55d
SOCKET_NETWORK socket1753  80
driver55a
SOCKET_NETWORK socket1754  80
driver55b
SOCKET_NETWORK socket1755  80
driver55c
SOCKET_NETWORK socket1756  80
driver55d
SOCKET_NETWORK socket1757  80
driver55a
SOCKET_NETWORK socket1758  80
driver55b
SOCKET_NETWORK socket1759  80
driver55c
SOCKET_NETWORK socket1760  80
driver55d
SOCKET_NETWORK socket1761  80
driver56a
SOCKET_NETWORK socket1762  80
driver56b
SOCKET_NETWORK socket1763  80
driver56c
SOCKET_NETWORK socket1764  80
driver56d
SOCKET_NETWORK socket1765  80
driver56a
SOCKET_NETWORK socket1766  80
driver56b
SOCKET_NETWORK socket1767  80
driver56c
SOCKET_NETWORK socket1768  80
driver56d
SOCKET_NETWORK socket1769  80
driver56a
SOCKET_NETWORK socket1770  80
driver56b
SOCKET_NETWORK socket1771  80
driver56c
SOCKET_NETWORK socket1772  80
driver56d
SOCKET_NETWORK socket1773  80
driver56a
SOCKET_NETWORK socket1774  80
driver56b
SOCKET_NETWORK socket1775  80
driver56c
SOCKET_NETWORK socket1776  80
driver56d
SOCKET_NETWORK socket1777  80
driver56a
SOCKET_NETWORK socket1778  80
driver56b
SOCKET_NETWORK socket1779  80
driver56c
SOCKET_NETWORK socket1780  80
driver56d
SOCKET_NETWORK socket1781  80
driver56a
SOCKET_NETWORK socket1782  80
driver56b
SOCKET_NETWORK socket1783  80
driver56c
SOCKET_NETWORK socket1784  80
driver56d
SOCKET_NETWORK socket1785  80
driver56a
```

```
SOCKET_NETWORK socket1786  80
driver56b
SOCKET_NETWORK socket1787  80
driver56c
SOCKET_NETWORK socket1788  80
driver56d
SOCKET_NETWORK socket1789  80
driver56a
SOCKET_NETWORK socket1790  80
driver56b
SOCKET_NETWORK socket1791  80
driver56c
SOCKET_NETWORK socket1792  80
driver56d
#elif MASTER_NUM15
SOCKET_NETWORK socket1793  80
driver57a
SOCKET_NETWORK socket1794  80
driver57b
SOCKET_NETWORK socket1795  80
driver57c
SOCKET_NETWORK socket1796  80
driver57d
SOCKET_NETWORK socket1797  80
driver57a
SOCKET_NETWORK socket1798  80
driver57b
SOCKET_NETWORK socket1799  80
driver57c
SOCKET_NETWORK socket1800  80
driver57d
SOCKET_NETWORK socket1801  80
driver57a
SOCKET_NETWORK socket1802  80
driver57b
SOCKET_NETWORK socket1803  80
driver57c
SOCKET_NETWORK socket1804  80
driver57d
SOCKET_NETWORK socket1805  80
driver57a
SOCKET_NETWORK socket1806  80
driver57b
SOCKET_NETWORK socket1807  80
driver57c
SOCKET_NETWORK socket1808  80
driver57d
SOCKET_NETWORK socket1809  80
driver57a
SOCKET_NETWORK socket1810  80
driver57b
SOCKET_NETWORK socket1811  80
driver57c
SOCKET_NETWORK socket1812  80
driver57d
SOCKET_NETWORK socket1813  80
driver57a
SOCKET_NETWORK socket1814  80
driver57b
SOCKET_NETWORK socket1815  80
driver57c
SOCKET_NETWORK socket1816  80
driver57d
SOCKET_NETWORK socket1817  80
driver57a
SOCKET_NETWORK socket1818  80
driver57b
SOCKET_NETWORK socket1819  80
driver57c
SOCKET_NETWORK socket1820  80
driver57d
SOCKET_NETWORK socket1821  80
driver57a
SOCKET_NETWORK socket1822  80
driver57b
SOCKET_NETWORK socket1823  80
driver57c
SOCKET_NETWORK socket1824  80
driver57d
SOCKET_NETWORK socket1825  80
driver58a
SOCKET_NETWORK socket1826  80
driver58b
SOCKET_NETWORK socket1827  80
driver58c
SOCKET_NETWORK socket1828  80
driver58d
SOCKET_NETWORK socket1829  80
driver58a
SOCKET_NETWORK socket1830  80
driver58b
SOCKET_NETWORK socket1831  80
driver58c
SOCKET_NETWORK socket1832  80
driver58d
SOCKET_NETWORK socket1833  80
driver58a
SOCKET_NETWORK socket1834  80
driver58b
SOCKET_NETWORK socket1835  80
driver58c
SOCKET_NETWORK socket1836  80
driver58d
SOCKET_NETWORK socket1837  80
driver58a
SOCKET_NETWORK socket1838  80
driver58b
SOCKET_NETWORK socket1839  80
driver58c
SOCKET_NETWORK socket1840  80
driver58d
SOCKET_NETWORK socket1841  80
driver58a
SOCKET_NETWORK socket1842  80
driver58b
SOCKET_NETWORK socket1843  80
driver58c
SOCKET_NETWORK socket1844  80
driver58d
SOCKET_NETWORK socket1845  80
driver58a
SOCKET_NETWORK socket1846  80
driver58b
SOCKET_NETWORK socket1847  80
driver58c
SOCKET_NETWORK socket1848  80
driver58d
SOCKET_NETWORK socket1849  80
driver58a
SOCKET_NETWORK socket1850  80
driver58b
SOCKET_NETWORK socket1851  80
driver58c
SOCKET_NETWORK socket1852  80
driver58d
SOCKET_NETWORK socket1853  80
driver58a
SOCKET_NETWORK socket1854  80
driver58b
SOCKET_NETWORK socket1855  80
driver58c
SOCKET_NETWORK socket1856  80
driver58d
SOCKET_NETWORK socket1857  80
driver59a
SOCKET_NETWORK socket1858  80
driver59b
SOCKET_NETWORK socket1859  80
driver59c
SOCKET_NETWORK socket1860  80
driver59d
SOCKET_NETWORK socket1861  80
driver59a
SOCKET_NETWORK socket1862  80
driver59b
SOCKET_NETWORK socket1863  80
driver59c
SOCKET_NETWORK socket1864  80
driver59d
SOCKET_NETWORK socket1865  80
driver59a
SOCKET_NETWORK socket1866  80
driver59b
SOCKET_NETWORK socket1867  80
driver59c
SOCKET_NETWORK socket1868  80
driver59d
SOCKET_NETWORK socket1869  80
driver59a
SOCKET_NETWORK socket1870  80
driver59b
SOCKET_NETWORK socket1871  80
driver59c
SOCKET_NETWORK socket1872  80
driver59d
SOCKET_NETWORK socket1873  80
driver59a
SOCKET_NETWORK socket1874  80
driver59b
SOCKET_NETWORK socket1875  80
driver59c
SOCKET_NETWORK socket1876  80
driver59d
SOCKET_NETWORK socket1877  80
driver59a
SOCKET_NETWORK socket1878  80
driver59b
SOCKET_NETWORK socket1879  80
driver59c
SOCKET_NETWORK socket1880  80
driver59d
SOCKET_NETWORK socket1881  80
driver59a
SOCKET_NETWORK socket1882  80
driver59b
SOCKET_NETWORK socket1883  80
driver59c
SOCKET_NETWORK socket1884  80
driver59d
SOCKET_NETWORK socket1885  80
driver59a
SOCKET_NETWORK socket1886  80
driver59b
SOCKET_NETWORK socket1887  80
driver59c
SOCKET_NETWORK socket1888  80
driver59d
SOCKET_NETWORK socket1889  80
driver60a
SOCKET_NETWORK socket1890  80
driver60b
SOCKET_NETWORK socket1891  80
driver60c
SOCKET_NETWORK socket1892  80
driver60d
SOCKET_NETWORK socket1893  80
driver60a
SOCKET_NETWORK socket1894  80
driver60b
SOCKET_NETWORK socket1895  80
driver60c
SOCKET_NETWORK socket1896  80
driver60d
SOCKET_NETWORK socket1897  80
driver60a
SOCKET_NETWORK socket1898  80
driver60b
SOCKET_NETWORK socket1899  80
driver60c
SOCKET_NETWORK socket1900  80
driver60d
SOCKET_NETWORK socket1901  80
driver60a
SOCKET_NETWORK socket1902  80
driver60b
SOCKET_NETWORK socket1903  80
driver60c
SOCKET_NETWORK socket1904  80
driver60d
SOCKET_NETWORK socket1905  80
driver60a
SOCKET_NETWORK socket1906  80
driver60b
SOCKET_NETWORK socket1907  80
driver60c
SOCKET_NETWORK socket1908  80
driver60d
```

```
SOCKET_NETWORK socket1909  80
driver60a
SOCKET_NETWORK socket1910  80
driver60b
SOCKET_NETWORK socket1911  80
driver60c
SOCKET_NETWORK socket1912  80
driver60d
SOCKET_NETWORK socket1913  80
driver60a
SOCKET_NETWORK socket1914  80
driver60b
SOCKET_NETWORK socket1915  80
driver60c
SOCKET_NETWORK socket1916  80
driver60d
SOCKET_NETWORK socket1917  80
driver60a
SOCKET_NETWORK socket1918  80
driver60b
SOCKET_NETWORK socket1919  80
driver60c
SOCKET_NETWORK socket1920  80
driver60d
#elif MASTER_NUM16
SOCKET_NETWORK socket1921  80
driver61a
SOCKET_NETWORK socket1922  80
driver61b
SOCKET_NETWORK socket1923  80
driver61c
SOCKET_NETWORK socket1924  80
driver61d
SOCKET_NETWORK socket1925  80
driver61a
SOCKET_NETWORK socket1926  80
driver61b
SOCKET_NETWORK socket1927  80
driver61c
SOCKET_NETWORK socket1928  80
driver61d
SOCKET_NETWORK socket1929  80
driver61a
SOCKET_NETWORK socket1930  80
driver61b
SOCKET_NETWORK socket1931  80
driver61c
SOCKET_NETWORK socket1932  80
driver61d
SOCKET_NETWORK socket1933  80
driver61a
SOCKET_NETWORK socket1934  80
driver61b
SOCKET_NETWORK socket1935  80
driver61c
SOCKET_NETWORK socket1936  80
driver61d
SOCKET_NETWORK socket1937  80
driver61a
SOCKET_NETWORK socket1938  80
driver61b
SOCKET_NETWORK socket1939  80
driver61c
SOCKET_NETWORK socket1940  80
driver61d
SOCKET_NETWORK socket1941  80
driver61a
SOCKET_NETWORK socket1942  80
driver61b
SOCKET_NETWORK socket1943  80
driver61c
SOCKET_NETWORK socket1944  80
driver61d
SOCKET_NETWORK socket1945  80
driver61a
SOCKET_NETWORK socket1946  80
driver61b
SOCKET_NETWORK socket1947  80
driver61c
SOCKET_NETWORK socket1948  80
driver61d
SOCKET_NETWORK socket1949  80
driver61a
SOCKET_NETWORK socket1950  80
driver61b
SOCKET_NETWORK socket1951  80
driver61c
SOCKET_NETWORK socket1952  80
driver61d
SOCKET_NETWORK socket1953  80
driver62a
SOCKET_NETWORK socket1954  80
driver62b
SOCKET_NETWORK socket1955  80
driver62c
SOCKET_NETWORK socket1956  80
driver62d
SOCKET_NETWORK socket1957  80
driver62a
SOCKET_NETWORK socket1958  80
driver62b
SOCKET_NETWORK socket1959  80
driver62c
SOCKET_NETWORK socket1960  80
driver62d
SOCKET_NETWORK socket1961  80
driver62a
SOCKET_NETWORK socket1962  80
driver62b
SOCKET_NETWORK socket1963  80
driver62c
SOCKET_NETWORK socket1964  80
driver62d
SOCKET_NETWORK socket1965  80
driver62a
SOCKET_NETWORK socket1966  80
driver62b
SOCKET_NETWORK socket1967  80
driver62c
SOCKET_NETWORK socket1968  80
driver62d
SOCKET_NETWORK socket1969  80
driver62a
SOCKET_NETWORK socket1970  80
driver62b
SOCKET_NETWORK socket1971  80
driver62c
SOCKET_NETWORK socket1972  80
driver62d
SOCKET_NETWORK socket1973  80
driver62a
SOCKET_NETWORK socket1974  80
driver62b
SOCKET_NETWORK socket1975  80
driver62c
SOCKET_NETWORK socket1976  80
driver62d
SOCKET_NETWORK socket1977  80
driver62a
SOCKET_NETWORK socket1978  80
driver62b
SOCKET_NETWORK socket1979  80
driver62c
SOCKET_NETWORK socket1980  80
driver62d
SOCKET_NETWORK socket1981  80
driver62a
SOCKET_NETWORK socket1982  80
driver62b
SOCKET_NETWORK socket1983  80
driver62c
SOCKET_NETWORK socket1984  80
driver62d
SOCKET_NETWORK socket1985  80
driver63a
SOCKET_NETWORK socket1986  80
driver63b
SOCKET_NETWORK socket1987  80
driver63c
SOCKET_NETWORK socket1988  80
driver63d
SOCKET_NETWORK socket1989  80
driver63a
SOCKET_NETWORK socket1990  80
driver63b
SOCKET_NETWORK socket1991  80
driver63c
SOCKET_NETWORK socket1992  80
driver63d
SOCKET_NETWORK socket1993  80
driver63a
SOCKET_NETWORK socket1994  80
driver63b
SOCKET_NETWORK socket1995  80
driver63c
SOCKET_NETWORK socket1996  80
driver63d
SOCKET_NETWORK socket1997  80
driver63a
SOCKET_NETWORK socket1998  80
driver63b
SOCKET_NETWORK socket1999  80
driver63c
SOCKET_NETWORK socket2000  80
driver63d
SOCKET_NETWORK socket2001  80
driver63a
SOCKET_NETWORK socket2002  80
driver63b
SOCKET_NETWORK socket2003  80
driver63c
SOCKET_NETWORK socket2004  80
driver63d
SOCKET_NETWORK socket2005  80
driver63a
SOCKET_NETWORK socket2006  80
driver63b
SOCKET_NETWORK socket2007  80
driver63c
SOCKET_NETWORK socket2008  80
driver63d
SOCKET_NETWORK socket2009  80
driver63a
SOCKET_NETWORK socket2010  80
driver63b
SOCKET_NETWORK socket2011  80
driver63c
SOCKET_NETWORK socket2012  80
driver63d
SOCKET_NETWORK socket2013  80
driver63a
SOCKET_NETWORK socket2014  80
driver63b
SOCKET_NETWORK socket2015  80
driver63c
SOCKET_NETWORK socket2016  80
driver63d
SOCKET_NETWORK socket2017  80
driver64a
SOCKET_NETWORK socket2018  80
driver64b
SOCKET_NETWORK socket2019  80
driver64c
SOCKET_NETWORK socket2020  80
driver64d
SOCKET_NETWORK socket2021  80
driver64a
SOCKET_NETWORK socket2022  80
driver64b
SOCKET_NETWORK socket2023  80
driver64c
SOCKET_NETWORK socket2024  80
driver64d
SOCKET_NETWORK socket2025  80
driver64a
SOCKET_NETWORK socket2026  80
driver64b
SOCKET_NETWORK socket2027  80
driver64c
SOCKET_NETWORK socket2028  80
driver64d
SOCKET_NETWORK socket2029  80
driver64a
SOCKET_NETWORK socket2030  80
driver64b
SOCKET_NETWORK socket2031  80
driver64c
```

```
SOCKET_NETWORK socket2032  80
driver64d
SOCKET_NETWORK socket2033  80
driver64a
SOCKET_NETWORK socket2034  80
driver64b
SOCKET_NETWORK socket2035  80
driver64c
SOCKET_NETWORK socket2036  80
driver64d
SOCKET_NETWORK socket2037  80
driver64a
SOCKET_NETWORK socket2038  80
driver64b
SOCKET_NETWORK socket2039  80
driver64c
SOCKET_NETWORK socket2040  80
driver64d
SOCKET_NETWORK socket2041  80
driver64a
SOCKET_NETWORK socket2042  80
driver64b
SOCKET_NETWORK socket2043  80
driver64c
SOCKET_NETWORK socket2044  80
driver64d
SOCKET_NETWORK socket2045  80
driver64a
SOCKET_NETWORK socket2046  80
driver64b
SOCKET_NETWORK socket2047  80
driver64c
SOCKET_NETWORK socket2048  80
driver64d
#elif MASTER_NUM17
SOCKET_NETWORK socket2049  80
driver65a
SOCKET_NETWORK socket2050  80
driver65b
SOCKET_NETWORK socket2051  80
driver65c
SOCKET_NETWORK socket2052  80
driver65d
SOCKET_NETWORK socket2053  80
driver65a
SOCKET_NETWORK socket2054  80
driver65b
SOCKET_NETWORK socket2055  80
driver65c
SOCKET_NETWORK socket2056  80
driver65d
SOCKET_NETWORK socket2057  80
driver65a
SOCKET_NETWORK socket2058  80
driver65b
SOCKET_NETWORK socket2059  80
driver65c
SOCKET_NETWORK socket2060  80
driver65d
SOCKET_NETWORK socket2061  80
driver65a
SOCKET_NETWORK socket2062  80
driver65b
SOCKET_NETWORK socket2063  80
driver65c
SOCKET_NETWORK socket2064  80
driver65d
SOCKET_NETWORK socket2065  80
driver65a
SOCKET_NETWORK socket2066  80
driver65b
SOCKET_NETWORK socket2067  80
driver65c
SOCKET_NETWORK socket2068  80
driver65d
SOCKET_NETWORK socket2069  80
driver65a
SOCKET_NETWORK socket2070  80
driver65b
SOCKET_NETWORK socket2071  80
driver65c
SOCKET_NETWORK socket2072  80
driver65d
SOCKET_NETWORK socket2073  80
driver65a
SOCKET_NETWORK socket2074  80
driver65b
SOCKET_NETWORK socket2075  80
driver65c
SOCKET_NETWORK socket2076  80
driver65d
SOCKET_NETWORK socket2077  80
driver65a
SOCKET_NETWORK socket2078  80
driver65b
SOCKET_NETWORK socket2079  80
driver65c
SOCKET_NETWORK socket2080  80
driver65d
SOCKET_NETWORK socket2081  80
driver66a
SOCKET_NETWORK socket2082  80
driver66b
SOCKET_NETWORK socket2083  80
driver66c
SOCKET_NETWORK socket2084  80
driver66d
SOCKET_NETWORK socket2085  80
driver66a
SOCKET_NETWORK socket2086  80
driver66b
SOCKET_NETWORK socket2087  80
driver66c
SOCKET_NETWORK socket2088  80
driver66d
SOCKET_NETWORK socket2089  80
driver66a
SOCKET_NETWORK socket2090  80
driver66b
SOCKET_NETWORK socket2091  80
driver66c
SOCKET_NETWORK socket2092  80
driver66d
SOCKET_NETWORK socket2093  80
driver66a
SOCKET_NETWORK socket2094  80
driver66b
SOCKET_NETWORK socket2095  80
driver66c
SOCKET_NETWORK socket2096  80
driver66d
SOCKET_NETWORK socket2097  80
driver66a
SOCKET_NETWORK socket2098  80
driver66b
SOCKET_NETWORK socket2099  80
driver66c
SOCKET_NETWORK socket2100  80
driver66d
SOCKET_NETWORK socket2101  80
driver66a
SOCKET_NETWORK socket2102  80
driver66b
SOCKET_NETWORK socket2103  80
driver66c
SOCKET_NETWORK socket2104  80
driver66d
SOCKET_NETWORK socket2105  80
driver66a
SOCKET_NETWORK socket2106  80
driver66b
SOCKET_NETWORK socket2107  80
driver66c
SOCKET_NETWORK socket2108  80
driver66d
SOCKET_NETWORK socket2109  80
driver66a
SOCKET_NETWORK socket2110  80
driver66b
SOCKET_NETWORK socket2111  80
driver66c
SOCKET_NETWORK socket2112  80
driver66d
SOCKET_NETWORK socket2113  80
driver67a
SOCKET_NETWORK socket2114  80
driver67b
SOCKET_NETWORK socket2115  80
driver67c
SOCKET_NETWORK socket2116  80
driver67d
SOCKET_NETWORK socket2117  80
driver67a
SOCKET_NETWORK socket2118  80
driver67b
SOCKET_NETWORK socket2119  80
driver67c
SOCKET_NETWORK socket2120  80
driver67d
SOCKET_NETWORK socket2121  80
driver67a
SOCKET_NETWORK socket2122  80
driver67b
SOCKET_NETWORK socket2123  80
driver67c
SOCKET_NETWORK socket2124  80
driver67d
SOCKET_NETWORK socket2125  80
driver67a
SOCKET_NETWORK socket2126  80
driver67b
SOCKET_NETWORK socket2127  80
driver67c
SOCKET_NETWORK socket2128  80
driver67d
SOCKET_NETWORK socket2129  80
driver67a
SOCKET_NETWORK socket2130  80
driver67b
SOCKET_NETWORK socket2131  80
driver67c
SOCKET_NETWORK socket2132  80
driver67d
SOCKET_NETWORK socket2133  80
driver67a
SOCKET_NETWORK socket2134  80
driver67b
SOCKET_NETWORK socket2135  80
driver67c
SOCKET_NETWORK socket2136  80
driver67d
SOCKET_NETWORK socket2137  80
driver67a
SOCKET_NETWORK socket2138  80
driver67b
SOCKET_NETWORK socket2139  80
driver67c
SOCKET_NETWORK socket2140  80
driver67d
SOCKET_NETWORK socket2141  80
driver67a
SOCKET_NETWORK socket2142  80
driver67b
SOCKET_NETWORK socket2143  80
driver67c
SOCKET_NETWORK socket2144  80
driver67d
SOCKET_NETWORK socket2145  80
driver68a
SOCKET_NETWORK socket2146  80
driver68b
SOCKET_NETWORK socket2147  80
driver68c
SOCKET_NETWORK socket2148  80
driver68d
SOCKET_NETWORK socket2149  80
driver68a
SOCKET_NETWORK socket2150  80
driver68b
SOCKET_NETWORK socket2151  80
driver68c
SOCKET_NETWORK socket2152  80
driver68d
SOCKET_NETWORK socket2153  80
driver68a
SOCKET_NETWORK socket2154  80
driver68b
```

```
SOCKET_NETWORK socket2155  80
driver68c
SOCKET_NETWORK socket2156  80
driver68d
SOCKET_NETWORK socket2157  80
driver68a
SOCKET_NETWORK socket2158  80
driver68b
SOCKET_NETWORK socket2159  80
driver68c
SOCKET_NETWORK socket2160  80
driver68d
SOCKET_NETWORK socket2161  80
driver68a
SOCKET_NETWORK socket2162  80
driver68b
SOCKET_NETWORK socket2163  80
driver68c
SOCKET_NETWORK socket2164  80
driver68d
SOCKET_NETWORK socket2165  80
driver68a
SOCKET_NETWORK socket2166  80
driver68b
SOCKET_NETWORK socket2167  80
driver68c
SOCKET_NETWORK socket2168  80
driver68d
SOCKET_NETWORK socket2169  80
driver68a
SOCKET_NETWORK socket2170  80
driver68b
SOCKET_NETWORK socket2171  80
driver68c
SOCKET_NETWORK socket2172  80
driver68d
SOCKET_NETWORK socket2173  80
driver68a
SOCKET_NETWORK socket2174  80
driver68b
SOCKET_NETWORK socket2175  80
driver68c
SOCKET_NETWORK socket2176  80
driver68d
#elif MASTER_NUM18
SOCKET_NETWORK socket2177  80
driver69a
SOCKET_NETWORK socket2178  80
driver69b
SOCKET_NETWORK socket2179  80
driver69c
SOCKET_NETWORK socket2180  80
driver69d
SOCKET_NETWORK socket2181  80
driver69a
SOCKET_NETWORK socket2182  80
driver69b
SOCKET_NETWORK socket2183  80
driver69c
SOCKET_NETWORK socket2184  80
driver69d
SOCKET_NETWORK socket2185  80
driver69a
SOCKET_NETWORK socket2186  80
driver69b
SOCKET_NETWORK socket2187  80
driver69c
SOCKET_NETWORK socket2188  80
driver69d
SOCKET_NETWORK socket2189  80
driver69a
SOCKET_NETWORK socket2190  80
driver69b
SOCKET_NETWORK socket2191  80
driver69c
SOCKET_NETWORK socket2192  80
driver69d
SOCKET_NETWORK socket2193  80
driver69a
SOCKET_NETWORK socket2194  80
driver69b
SOCKET_NETWORK socket2195  80
driver69c
SOCKET_NETWORK socket2196  80
driver69d
SOCKET_NETWORK socket2197  80
driver69a
SOCKET_NETWORK socket2198  80
driver69b
SOCKET_NETWORK socket2199  80
driver69c
SOCKET_NETWORK socket2200  80
driver69d
SOCKET_NETWORK socket2201  80
driver69a
SOCKET_NETWORK socket2202  80
driver69b
SOCKET_NETWORK socket2203  80
driver69c
SOCKET_NETWORK socket2204  80
driver69d
SOCKET_NETWORK socket2205  80
driver69a
SOCKET_NETWORK socket2206  80
driver69b
SOCKET_NETWORK socket2207  80
driver69c
SOCKET_NETWORK socket2208  80
driver69d
SOCKET_NETWORK socket2209  80
driver70a
SOCKET_NETWORK socket2210  80
driver70b
SOCKET_NETWORK socket2211  80
driver70c
SOCKET_NETWORK socket2212  80
driver70d
SOCKET_NETWORK socket2213  80
driver70a
SOCKET_NETWORK socket2214  80
driver70b
SOCKET_NETWORK socket2215  80
driver70c
SOCKET_NETWORK socket2216  80
driver70d
SOCKET_NETWORK socket2217  80
driver70a
SOCKET_NETWORK socket2218  80
driver70b
SOCKET_NETWORK socket2219  80
driver70c
SOCKET_NETWORK socket2220  80
driver70d
SOCKET_NETWORK socket2221  80
driver70a
SOCKET_NETWORK socket2222  80
driver70b
SOCKET_NETWORK socket2223  80
driver70c
SOCKET_NETWORK socket2224  80
driver70d
SOCKET_NETWORK socket2225  80
driver70a
SOCKET_NETWORK socket2226  80
driver70b
SOCKET_NETWORK socket2227  80
driver70c
SOCKET_NETWORK socket2228  80
driver70d
SOCKET_NETWORK socket2229  80
driver70a
SOCKET_NETWORK socket2230  80
driver70b
SOCKET_NETWORK socket2231  80
driver70c
SOCKET_NETWORK socket2232  80
driver70d
SOCKET_NETWORK socket2233  80
driver70a
SOCKET_NETWORK socket2234  80
driver70b
SOCKET_NETWORK socket2235  80
driver70c
SOCKET_NETWORK socket2236  80
driver70d
SOCKET_NETWORK socket2237  80
driver70a
SOCKET_NETWORK socket2238  80
driver70b
SOCKET_NETWORK socket2239  80
driver70c
SOCKET_NETWORK socket2240  80
driver70d
SOCKET_NETWORK socket2241  80
driver71a
SOCKET_NETWORK socket2242  80
driver71b
SOCKET_NETWORK socket2243  80
driver71c
SOCKET_NETWORK socket2244  80
driver71d
SOCKET_NETWORK socket2245  80
driver71a
SOCKET_NETWORK socket2246  80
driver71b
SOCKET_NETWORK socket2247  80
driver71c
SOCKET_NETWORK socket2248  80
driver71d
SOCKET_NETWORK socket2249  80
driver71a
SOCKET_NETWORK socket2250  80
driver71b
SOCKET_NETWORK socket2251  80
driver71c
SOCKET_NETWORK socket2252  80
driver71d
SOCKET_NETWORK socket2253  80
driver71a
SOCKET_NETWORK socket2254  80
driver71b
SOCKET_NETWORK socket2255  80
driver71c
SOCKET_NETWORK socket2256  80
driver71d
SOCKET_NETWORK socket2257  80
driver71a
SOCKET_NETWORK socket2258  80
driver71b
SOCKET_NETWORK socket2259  80
driver71c
SOCKET_NETWORK socket2260  80
driver71d
SOCKET_NETWORK socket2261  80
driver71a
SOCKET_NETWORK socket2262  80
driver71b
SOCKET_NETWORK socket2263  80
driver71c
SOCKET_NETWORK socket2264  80
driver71d
SOCKET_NETWORK socket2265  80
driver71a
SOCKET_NETWORK socket2266  80
driver71b
SOCKET_NETWORK socket2267  80
driver71c
SOCKET_NETWORK socket2268  80
driver71d
SOCKET_NETWORK socket2269  80
driver71a
SOCKET_NETWORK socket2270  80
driver71b
SOCKET_NETWORK socket2271  80
driver71c
SOCKET_NETWORK socket2272  80
driver71d
SOCKET_NETWORK socket2273  80
driver72a
SOCKET_NETWORK socket2274  80
driver72b
SOCKET_NETWORK socket2275  80
driver72c
SOCKET_NETWORK socket2276  80
driver72d
SOCKET_NETWORK socket2277  80
driver72a
```

```
SOCKET_NETWORK socket2278 80
driver72b
SOCKET_NETWORK socket2279 80
driver72c
SOCKET_NETWORK socket2280 80
driver72d
SOCKET_NETWORK socket2281 80
driver72a
SOCKET_NETWORK socket2282 80
driver72b
SOCKET_NETWORK socket2283 80
driver72c
SOCKET_NETWORK socket2284 80
driver72d
SOCKET_NETWORK socket2285 80
driver72a
SOCKET_NETWORK socket2286 80
driver72b
SOCKET_NETWORK socket2287 80
driver72c
SOCKET_NETWORK socket2288 80
driver72d
SOCKET_NETWORK socket2289 80
driver72a
SOCKET_NETWORK socket2290 80
driver72b
SOCKET_NETWORK socket2291 80
driver72c
SOCKET_NETWORK socket2292 80
driver72d
SOCKET_NETWORK socket2293 80
driver72a
SOCKET_NETWORK socket2294 80
driver72b
SOCKET_NETWORK socket2295 80
driver72c
SOCKET_NETWORK socket2296 80
driver72d
SOCKET_NETWORK socket2297 80
driver72a
SOCKET_NETWORK socket2298 80
driver72b
SOCKET_NETWORK socket2299 80
driver72c
SOCKET_NETWORK socket2300 80
driver72d
SOCKET_NETWORK socket2301 80
driver72a
SOCKET_NETWORK socket2302 80
driver72b
SOCKET_NETWORK socket2303 80
driver72c
SOCKET_NETWORK socket2304 80
driver72d
#elif MASTER_NUM19
SOCKET_NETWORK socket2305 80
driver73a
SOCKET_NETWORK socket2306 80
driver73b
SOCKET_NETWORK socket2307 80
driver73c
SOCKET_NETWORK socket2308 80
driver73d
SOCKET_NETWORK socket2309 80
driver73a
SOCKET_NETWORK socket2310 80
driver73b
SOCKET_NETWORK socket2311 80
driver73c
SOCKET_NETWORK socket2312 80
driver73d
SOCKET_NETWORK socket2313 80
driver73a
SOCKET_NETWORK socket2314 80
driver73b
SOCKET_NETWORK socket2315 80
driver73c
SOCKET_NETWORK socket2316 80
driver73d
SOCKET_NETWORK socket2317 80
driver73a
SOCKET_NETWORK socket2318 80
driver73b
SOCKET_NETWORK socket2319 80
driver73c
SOCKET_NETWORK socket2320 80
driver73d
SOCKET_NETWORK socket2321 80
driver73a
SOCKET_NETWORK socket2322 80
driver73b
SOCKET_NETWORK socket2323 80
driver73c
SOCKET_NETWORK socket2324 80
driver73d
SOCKET_NETWORK socket2325 80
driver73a
SOCKET_NETWORK socket2326 80
driver73b
SOCKET_NETWORK socket2327 80
driver73c
SOCKET_NETWORK socket2328 80
driver73d
SOCKET_NETWORK socket2329 80
driver73a
SOCKET_NETWORK socket2330 80
driver73b
SOCKET_NETWORK socket2331 80
driver73c
SOCKET_NETWORK socket2332 80
driver73d
SOCKET_NETWORK socket2333 80
driver73a
SOCKET_NETWORK socket2334 80
driver73b
SOCKET_NETWORK socket2335 80
driver73c
SOCKET_NETWORK socket2336 80
driver73d
SOCKET_NETWORK socket2337 80
driver74a
SOCKET_NETWORK socket2338 80
driver74b
SOCKET_NETWORK socket2339 80
driver74c
SOCKET_NETWORK socket2340 80
driver74d
SOCKET_NETWORK socket2341 80
driver74a
SOCKET_NETWORK socket2342 80
driver74b
SOCKET_NETWORK socket2343 80
driver74c
SOCKET_NETWORK socket2344 80
driver74d
SOCKET_NETWORK socket2345 80
driver74a
SOCKET_NETWORK socket2346 80
driver74b
SOCKET_NETWORK socket2347 80
driver74c
SOCKET_NETWORK socket2348 80
driver74d
SOCKET_NETWORK socket2349 80
driver74a
SOCKET_NETWORK socket2350 80
driver74b
SOCKET_NETWORK socket2351 80
driver74c
SOCKET_NETWORK socket2352 80
driver74d
SOCKET_NETWORK socket2353 80
driver74a
SOCKET_NETWORK socket2354 80
driver74b
SOCKET_NETWORK socket2355 80
driver74c
SOCKET_NETWORK socket2356 80
driver74d
SOCKET_NETWORK socket2357 80
driver74a
SOCKET_NETWORK socket2358 80
driver74b
SOCKET_NETWORK socket2359 80
driver74c
SOCKET_NETWORK socket2360 80
driver74d
SOCKET_NETWORK socket2361 80
driver74a
SOCKET_NETWORK socket2362 80
driver74b
SOCKET_NETWORK socket2363 80
driver74c
SOCKET_NETWORK socket2364 80
driver74d
SOCKET_NETWORK socket2365 80
driver74a
SOCKET_NETWORK socket2366 80
driver74b
SOCKET_NETWORK socket2367 80
driver74c
SOCKET_NETWORK socket2368 80
driver74d
SOCKET_NETWORK socket2369 80
driver75a
SOCKET_NETWORK socket2370 80
driver75b
SOCKET_NETWORK socket2371 80
driver75c
SOCKET_NETWORK socket2372 80
driver75d
SOCKET_NETWORK socket2373 80
driver75a
SOCKET_NETWORK socket2374 80
driver75b
SOCKET_NETWORK socket2375 80
driver75c
SOCKET_NETWORK socket2376 80
driver75d
SOCKET_NETWORK socket2377 80
driver75a
SOCKET_NETWORK socket2378 80
driver75b
SOCKET_NETWORK socket2379 80
driver75c
SOCKET_NETWORK socket2380 80
driver75d
SOCKET_NETWORK socket2381 80
driver75a
SOCKET_NETWORK socket2382 80
driver75b
SOCKET_NETWORK socket2383 80
driver75c
SOCKET_NETWORK socket2384 80
driver75d
SOCKET_NETWORK socket2385 80
driver75a
SOCKET_NETWORK socket2386 80
driver75b
SOCKET_NETWORK socket2387 80
driver75c
SOCKET_NETWORK socket2388 80
driver75d
SOCKET_NETWORK socket2389 80
driver75a
SOCKET_NETWORK socket2390 80
driver75b
SOCKET_NETWORK socket2391 80
driver75c
SOCKET_NETWORK socket2392 80
driver75d
SOCKET_NETWORK socket2393 80
driver75a
SOCKET_NETWORK socket2394 80
driver75b
SOCKET_NETWORK socket2395 80
driver75c
SOCKET_NETWORK socket2396 80
driver75d
SOCKET_NETWORK socket2397 80
driver75a
SOCKET_NETWORK socket2398 80
driver75b
SOCKET_NETWORK socket2399 80
driver75c
SOCKET_NETWORK socket2400 80
driver75d
```

```
SOCKET_NETWORK socket2401  80
driver76a
SOCKET_NETWORK socket2402  80
driver76b
SOCKET_NETWORK socket2403  80
driver76c
SOCKET_NETWORK socket2404  80
driver76d
SOCKET_NETWORK socket2405  80
driver76a
SOCKET_NETWORK socket2406  80
driver76b
SOCKET_NETWORK socket2407  80
driver76c
SOCKET_NETWORK socket2408  80
driver76d
SOCKET_NETWORK socket2409  80
driver76a
SOCKET_NETWORK socket2410  80
driver76b
SOCKET_NETWORK socket2411  80
driver76c
SOCKET_NETWORK socket2412  80
driver76d
SOCKET_NETWORK socket2413  80
driver76a
SOCKET_NETWORK socket2414  80
driver76b
SOCKET_NETWORK socket2415  80
driver76c
SOCKET_NETWORK socket2416  80
driver76d
SOCKET_NETWORK socket2417  80
driver76a
SOCKET_NETWORK socket2418  80
driver76b
SOCKET_NETWORK socket2419  80
driver76c
SOCKET_NETWORK socket2420  80
driver76d
SOCKET_NETWORK socket2421  80
driver76a
SOCKET_NETWORK socket2422  80
driver76b
SOCKET_NETWORK socket2423  80
driver76c
SOCKET_NETWORK socket2424  80
driver76d
SOCKET_NETWORK socket2425  80
driver76a
SOCKET_NETWORK socket2426  80
driver76b
SOCKET_NETWORK socket2427  80
driver76c
SOCKET_NETWORK socket2428  80
driver76d
SOCKET_NETWORK socket2429  80
driver76a
SOCKET_NETWORK socket2430  80
driver76b
SOCKET_NETWORK socket2431  80
driver76c
SOCKET_NETWORK socket2432  80
driver76d
#elif MASTER_NUM20
SOCKET_NETWORK socket2433  80
driver77a
SOCKET_NETWORK socket2434  80
driver77b
SOCKET_NETWORK socket2435  80
driver77c
SOCKET_NETWORK socket2436  80
driver77d
SOCKET_NETWORK socket2437  80
driver77a
SOCKET_NETWORK socket2438  80
driver77b
SOCKET_NETWORK socket2439  80
driver77c
SOCKET_NETWORK socket2440  80
driver77d
SOCKET_NETWORK socket2441  80
driver77a
SOCKET_NETWORK socket2442  80
driver77b
SOCKET_NETWORK socket2443  80
driver77c
SOCKET_NETWORK socket2444  80
driver77d
SOCKET_NETWORK socket2445  80
driver77a
SOCKET_NETWORK socket2446  80
driver77b
SOCKET_NETWORK socket2447  80
driver77c
SOCKET_NETWORK socket2448  80
driver77d
SOCKET_NETWORK socket2449  80
driver77a
SOCKET_NETWORK socket2450  80
driver77b
SOCKET_NETWORK socket2451  80
driver77c
SOCKET_NETWORK socket2452  80
driver77d
SOCKET_NETWORK socket2453  80
driver77a
SOCKET_NETWORK socket2454  80
driver77b
SOCKET_NETWORK socket2455  80
driver77c
SOCKET_NETWORK socket2456  80
driver77d
SOCKET_NETWORK socket2457  80
driver77a
SOCKET_NETWORK socket2458  80
driver77b
SOCKET_NETWORK socket2459  80
driver77c
SOCKET_NETWORK socket2460  80
driver77d
SOCKET_NETWORK socket2461  80
driver77a
SOCKET_NETWORK socket2462  80
driver77b
SOCKET_NETWORK socket2463  80
driver77c
SOCKET_NETWORK socket2464  80
driver77d
SOCKET_NETWORK socket2465  80
driver78a
SOCKET_NETWORK socket2466  80
driver78b
SOCKET_NETWORK socket2467  80
driver78c
SOCKET_NETWORK socket2468  80
driver78d
SOCKET_NETWORK socket2469  80
driver78a
SOCKET_NETWORK socket2470  80
driver78b
SOCKET_NETWORK socket2471  80
driver78c
SOCKET_NETWORK socket2472  80
driver78d
SOCKET_NETWORK socket2473  80
driver78a
SOCKET_NETWORK socket2474  80
driver78b
SOCKET_NETWORK socket2475  80
driver78c
SOCKET_NETWORK socket2476  80
driver78d
SOCKET_NETWORK socket2477  80
driver78a
SOCKET_NETWORK socket2478  80
driver78b
SOCKET_NETWORK socket2479  80
driver78c
SOCKET_NETWORK socket2480  80
driver78d
SOCKET_NETWORK socket2481  80
driver78a
SOCKET_NETWORK socket2482  80
driver78b
SOCKET_NETWORK socket2483  80
driver78c
SOCKET_NETWORK socket2484  80
driver78d
SOCKET_NETWORK socket2485  80
driver78a
SOCKET_NETWORK socket2486  80
driver78b
SOCKET_NETWORK socket2487  80
driver78c
SOCKET_NETWORK socket2488  80
driver78d
SOCKET_NETWORK socket2489  80
driver78a
SOCKET_NETWORK socket2490  80
driver78b
SOCKET_NETWORK socket2491  80
driver78c
SOCKET_NETWORK socket2492  80
driver78d
SOCKET_NETWORK socket2493  80
driver78a
SOCKET_NETWORK socket2494  80
driver78b
SOCKET_NETWORK socket2495  80
driver78c
SOCKET_NETWORK socket2496  80
driver78d
SOCKET_NETWORK socket2497  80
driver79a
SOCKET_NETWORK socket2498  80
driver79b
SOCKET_NETWORK socket2499  80
driver79c
SOCKET_NETWORK socket2500  80
driver79d
SOCKET_NETWORK socket2501  80
driver79a
SOCKET_NETWORK socket2502  80
driver79b
SOCKET_NETWORK socket2503  80
driver79c
SOCKET_NETWORK socket2504  80
driver79d
SOCKET_NETWORK socket2505  80
driver79a
SOCKET_NETWORK socket2506  80
driver79b
SOCKET_NETWORK socket2507  80
driver79c
SOCKET_NETWORK socket2508  80
driver79d
SOCKET_NETWORK socket2509  80
driver79a
SOCKET_NETWORK socket2510  80
driver79b
SOCKET_NETWORK socket2511  80
driver79c
SOCKET_NETWORK socket2512  80
driver79d
SOCKET_NETWORK socket2513  80
driver79a
SOCKET_NETWORK socket2514  80
driver79b
SOCKET_NETWORK socket2515  80
driver79c
SOCKET_NETWORK socket2516  80
driver79d
SOCKET_NETWORK socket2517  80
driver79a
SOCKET_NETWORK socket2518  80
driver79b
SOCKET_NETWORK socket2519  80
driver79c
SOCKET_NETWORK socket2520  80
driver79d
SOCKET_NETWORK socket2521  80
driver79a
SOCKET_NETWORK socket2522  80
driver79b
SOCKET_NETWORK socket2523  80
driver79c
```

```
SOCKET_NETWORK socket2524  80
driver79d
SOCKET_NETWORK socket2525  80
driver79a
SOCKET_NETWORK socket2526  80
driver79b
SOCKET_NETWORK socket2527  80
driver79c
SOCKET_NETWORK socket2528  80
driver79d
SOCKET_NETWORK socket2529  80
driver80a
SOCKET_NETWORK socket2530  80
driver80b
SOCKET_NETWORK socket2531  80
driver80c
SOCKET_NETWORK socket2532  80
driver80d
SOCKET_NETWORK socket2533  80
driver80a
SOCKET_NETWORK socket2534  80
driver80b
SOCKET_NETWORK socket2535  80
driver80c
SOCKET_NETWORK socket2536  80
driver80d
SOCKET_NETWORK socket2537  80
driver80a
SOCKET_NETWORK socket2538  80
driver80b
SOCKET_NETWORK socket2539  80
driver80c
SOCKET_NETWORK socket2540  80
driver80d
SOCKET_NETWORK socket2541  80
driver80a
SOCKET_NETWORK socket2542  80
driver80b
SOCKET_NETWORK socket2543  80
driver80c
SOCKET_NETWORK socket2544  80
driver80d
SOCKET_NETWORK socket2545  80
driver80a
SOCKET_NETWORK socket2546  80
driver80b
SOCKET_NETWORK socket2547  80
driver80c
SOCKET_NETWORK socket2548  80
driver80d
SOCKET_NETWORK socket2549  80
driver80a
SOCKET_NETWORK socket2550  80
driver80b
SOCKET_NETWORK socket2551  80
driver80c
SOCKET_NETWORK socket2552  80
driver80d
SOCKET_NETWORK socket2553  80
driver80a
SOCKET_NETWORK socket2554  80
driver80b
SOCKET_NETWORK socket2555  80
driver80c
SOCKET_NETWORK socket2556  80
driver80d
SOCKET_NETWORK socket2557  80
driver80a
SOCKET_NETWORK socket2558  80
driver80b
SOCKET_NETWORK socket2559  80
driver80c
SOCKET_NETWORK socket2560  80
driver80d
#elif MASTER_NUM21
SOCKET_NETWORK socket2561  80
driver81a
SOCKET_NETWORK socket2562  80
driver81b
SOCKET_NETWORK socket2563  80
driver81c
SOCKET_NETWORK socket2564  80
driver81d
SOCKET_NETWORK socket2565  80
driver81a
SOCKET_NETWORK socket2566  80
driver81b
SOCKET_NETWORK socket2567  80
driver81c
SOCKET_NETWORK socket2568  80
driver81d
SOCKET_NETWORK socket2569  80
driver81a
SOCKET_NETWORK socket2570  80
driver81b
SOCKET_NETWORK socket2571  80
driver81c
SOCKET_NETWORK socket2572  80
driver81d
SOCKET_NETWORK socket2573  80
driver81a
SOCKET_NETWORK socket2574  80
driver81b
SOCKET_NETWORK socket2575  80
driver81c
SOCKET_NETWORK socket2576  80
driver81d
SOCKET_NETWORK socket2577  80
driver81a
SOCKET_NETWORK socket2578  80
driver81b
SOCKET_NETWORK socket2579  80
driver81c
SOCKET_NETWORK socket2580  80
driver81d
SOCKET_NETWORK socket2581  80
driver81a
SOCKET_NETWORK socket2582  80
driver81b
SOCKET_NETWORK socket2583  80
driver81c
SOCKET_NETWORK socket2584  80
driver81d
SOCKET_NETWORK socket2585  80
driver81a
SOCKET_NETWORK socket2586  80
driver81b
SOCKET_NETWORK socket2587  80
driver81c
SOCKET_NETWORK socket2588  80
driver81d
SOCKET_NETWORK socket2589  80
driver81a
SOCKET_NETWORK socket2590  80
driver81b
SOCKET_NETWORK socket2591  80
driver81c
SOCKET_NETWORK socket2592  80
driver81d
SOCKET_NETWORK socket2593  80
driver82a
SOCKET_NETWORK socket2594  80
driver82b
SOCKET_NETWORK socket2595  80
driver82c
SOCKET_NETWORK socket2596  80
driver82d
SOCKET_NETWORK socket2597  80
driver82a
SOCKET_NETWORK socket2598  80
driver82b
SOCKET_NETWORK socket2599  80
driver82c
SOCKET_NETWORK socket2600  80
driver82d
SOCKET_NETWORK socket2601  80
driver82a
SOCKET_NETWORK socket2602  80
driver82b
SOCKET_NETWORK socket2603  80
driver82c
SOCKET_NETWORK socket2604  80
driver82d
SOCKET_NETWORK socket2605  80
driver82a
SOCKET_NETWORK socket2606  80
driver82b
SOCKET_NETWORK socket2607  80
driver82c
SOCKET_NETWORK socket2608  80
driver82d
SOCKET_NETWORK socket2609  80
driver82a
SOCKET_NETWORK socket2610  80
driver82b
SOCKET_NETWORK socket2611  80
driver82c
SOCKET_NETWORK socket2612  80
driver82d
SOCKET_NETWORK socket2613  80
driver82a
SOCKET_NETWORK socket2614  80
driver82b
SOCKET_NETWORK socket2615  80
driver82c
SOCKET_NETWORK socket2616  80
driver82d
SOCKET_NETWORK socket2617  80
driver82a
SOCKET_NETWORK socket2618  80
driver82b
SOCKET_NETWORK socket2619  80
driver82c
SOCKET_NETWORK socket2620  80
driver82d
SOCKET_NETWORK socket2621  80
driver82a
SOCKET_NETWORK socket2622  80
driver82b
SOCKET_NETWORK socket2623  80
driver82c
SOCKET_NETWORK socket2624  80
driver82d
SOCKET_NETWORK socket2625  80
driver83a
SOCKET_NETWORK socket2626  80
driver83b
SOCKET_NETWORK socket2627  80
driver83c
SOCKET_NETWORK socket2628  80
driver83d
SOCKET_NETWORK socket2629  80
driver83a
SOCKET_NETWORK socket2630  80
driver83b
SOCKET_NETWORK socket2631  80
driver83c
SOCKET_NETWORK socket2632  80
driver83d
SOCKET_NETWORK socket2633  80
driver83a
SOCKET_NETWORK socket2634  80
driver83b
SOCKET_NETWORK socket2635  80
driver83c
SOCKET_NETWORK socket2636  80
driver83d
SOCKET_NETWORK socket2637  80
driver83a
SOCKET_NETWORK socket2638  80
driver83b
SOCKET_NETWORK socket2639  80
driver83c
SOCKET_NETWORK socket2640  80
driver83d
SOCKET_NETWORK socket2641  80
driver83a
SOCKET_NETWORK socket2642  80
driver83b
SOCKET_NETWORK socket2643  80
driver83c
SOCKET_NETWORK socket2644  80
driver83d
SOCKET_NETWORK socket2645  80
driver83a
SOCKET_NETWORK socket2646  80
driver83b
```

```
SOCKET_NETWORK socket2647  80
driver83c
SOCKET_NETWORK socket2648  80
driver83d
SOCKET_NETWORK socket2649  80
driver83a
SOCKET_NETWORK socket2650  80
driver83b
SOCKET_NETWORK socket2651  80
driver83c
SOCKET_NETWORK socket2652  80
driver83d
SOCKET_NETWORK socket2653  80
driver83a
SOCKET_NETWORK socket2654  80
driver83b
SOCKET_NETWORK socket2655  80
driver83c
SOCKET_NETWORK socket2656  80
driver83d
SOCKET_NETWORK socket2657  80
driver84a
SOCKET_NETWORK socket2658  80
driver84b
SOCKET_NETWORK socket2659  80
driver84c
SOCKET_NETWORK socket2660  80
driver84d
SOCKET_NETWORK socket2661  80
driver84a
SOCKET_NETWORK socket2662  80
driver84b
SOCKET_NETWORK socket2663  80
driver84c
SOCKET_NETWORK socket2664  80
driver84d
SOCKET_NETWORK socket2665  80
driver84a
SOCKET_NETWORK socket2666  80
driver84b
SOCKET_NETWORK socket2667  80
driver84c
SOCKET_NETWORK socket2668  80
driver84d
SOCKET_NETWORK socket2669  80
driver84a
SOCKET_NETWORK socket2670  80
driver84b
SOCKET_NETWORK socket2671  80
driver84c
SOCKET_NETWORK socket2672  80
driver84d
SOCKET_NETWORK socket2673  80
driver84a
SOCKET_NETWORK socket2674  80
driver84b
SOCKET_NETWORK socket2675  80
driver84c
SOCKET_NETWORK socket2676  80
driver84d
SOCKET_NETWORK socket2677  80
driver84a
SOCKET_NETWORK socket2678  80
driver84b
SOCKET_NETWORK socket2679  80
driver84c
SOCKET_NETWORK socket2680  80
driver84d
SOCKET_NETWORK socket2681  80
driver84a
SOCKET_NETWORK socket2682  80
driver84b
SOCKET_NETWORK socket2683  80
driver84c
SOCKET_NETWORK socket2684  80
driver84d
SOCKET_NETWORK socket2685  80
driver84a
SOCKET_NETWORK socket2686  80
driver84b
SOCKET_NETWORK socket2687  80
driver84c
SOCKET_NETWORK socket2688  80
driver84d
#elif MASTER_NUM22
SOCKET_NETWORK socket2689  80
driver85a
SOCKET_NETWORK socket2690  80
driver85b
SOCKET_NETWORK socket2691  80
driver85c
SOCKET_NETWORK socket2692  80
driver85d
SOCKET_NETWORK socket2693  80
driver85a
SOCKET_NETWORK socket2694  80
driver85b
SOCKET_NETWORK socket2695  80
driver85c
SOCKET_NETWORK socket2696  80
driver85d
SOCKET_NETWORK socket2697  80
driver85a
SOCKET_NETWORK socket2698  80
driver85b
SOCKET_NETWORK socket2699  80
driver85c
SOCKET_NETWORK socket2700  80
driver85d
SOCKET_NETWORK socket2701  80
driver85a
SOCKET_NETWORK socket2702  80
driver85b
SOCKET_NETWORK socket2703  80
driver85c
SOCKET_NETWORK socket2704  80
driver85d
SOCKET_NETWORK socket2705  80
driver85a
SOCKET_NETWORK socket2706  80
driver85b
SOCKET_NETWORK socket2707  80
driver85c
SOCKET_NETWORK socket2708  80
driver85d
SOCKET_NETWORK socket2709  80
driver85a
SOCKET_NETWORK socket2710  80
driver85b
SOCKET_NETWORK socket2711  80
driver85c
SOCKET_NETWORK socket2712  80
driver85d
SOCKET_NETWORK socket2713  80
driver85a
SOCKET_NETWORK socket2714  80
driver85b
SOCKET_NETWORK socket2715  80
driver85c
SOCKET_NETWORK socket2716  80
driver85d
SOCKET_NETWORK socket2717  80
driver85a
SOCKET_NETWORK socket2718  80
driver85b
SOCKET_NETWORK socket2719  80
driver85c
SOCKET_NETWORK socket2720  80
driver85d
SOCKET_NETWORK socket2721  80
driver86a
SOCKET_NETWORK socket2722  80
driver86b
SOCKET_NETWORK socket2723  80
driver86c
SOCKET_NETWORK socket2724  80
driver86d
SOCKET_NETWORK socket2725  80
driver86a
SOCKET_NETWORK socket2726  80
driver86b
SOCKET_NETWORK socket2727  80
driver86c
SOCKET_NETWORK socket2728  80
driver86d
SOCKET_NETWORK socket2729  80
driver86a
SOCKET_NETWORK socket2730  80
driver86b
SOCKET_NETWORK socket2731  80
driver86c
SOCKET_NETWORK socket2732  80
driver86d
SOCKET_NETWORK socket2733  80
driver86a
SOCKET_NETWORK socket2734  80
driver86b
SOCKET_NETWORK socket2735  80
driver86c
SOCKET_NETWORK socket2736  80
driver86d
SOCKET_NETWORK socket2737  80
driver86a
SOCKET_NETWORK socket2738  80
driver86b
SOCKET_NETWORK socket2739  80
driver86c
SOCKET_NETWORK socket2740  80
driver86d
SOCKET_NETWORK socket2741  80
driver86a
SOCKET_NETWORK socket2742  80
driver86b
SOCKET_NETWORK socket2743  80
driver86c
SOCKET_NETWORK socket2744  80
driver86d
SOCKET_NETWORK socket2745  80
driver86a
SOCKET_NETWORK socket2746  80
driver86b
SOCKET_NETWORK socket2747  80
driver86c
SOCKET_NETWORK socket2748  80
driver86d
SOCKET_NETWORK socket2749  80
driver86a
SOCKET_NETWORK socket2750  80
driver86b
SOCKET_NETWORK socket2751  80
driver86c
SOCKET_NETWORK socket2752  80
driver86d
SOCKET_NETWORK socket2753  80
driver87a
SOCKET_NETWORK socket2754  80
driver87b
SOCKET_NETWORK socket2755  80
driver87c
SOCKET_NETWORK socket2756  80
driver87d
SOCKET_NETWORK socket2757  80
driver87a
SOCKET_NETWORK socket2758  80
driver87b
SOCKET_NETWORK socket2759  80
driver87c
SOCKET_NETWORK socket2760  80
driver87d
SOCKET_NETWORK socket2761  80
driver87a
SOCKET_NETWORK socket2762  80
driver87b
SOCKET_NETWORK socket2763  80
driver87c
SOCKET_NETWORK socket2764  80
driver87d
SOCKET_NETWORK socket2765  80
driver87a
SOCKET_NETWORK socket2766  80
driver87b
SOCKET_NETWORK socket2767  80
driver87c
SOCKET_NETWORK socket2768  80
driver87d
SOCKET_NETWORK socket2769  80
driver87a
```

```
SOCKET_NETWORK socket2770 80
driver87b
SOCKET_NETWORK socket2771 80
driver87c
SOCKET_NETWORK socket2772 80
driver87d
SOCKET_NETWORK socket2773 80
driver87a
SOCKET_NETWORK socket2774 80
driver87b
SOCKET_NETWORK socket2775 80
driver87c
SOCKET_NETWORK socket2776 80
driver87d
SOCKET_NETWORK socket2777 80
driver87a
SOCKET_NETWORK socket2778 80
driver87b
SOCKET_NETWORK socket2779 80
driver87c
SOCKET_NETWORK socket2780 80
driver87d
SOCKET_NETWORK socket2781 80
driver87a
SOCKET_NETWORK socket2782 80
driver87b
SOCKET_NETWORK socket2783 80
driver87c
SOCKET_NETWORK socket2784 80
driver87d
SOCKET_NETWORK socket2785 80
driver88a
SOCKET_NETWORK socket2786 80
driver88b
SOCKET_NETWORK socket2787 80
driver88c
SOCKET_NETWORK socket2788 80
driver88d
SOCKET_NETWORK socket2789 80
driver88a
SOCKET_NETWORK socket2790 80
driver88b
SOCKET_NETWORK socket2791 80
driver88c
SOCKET_NETWORK socket2792 80
driver88d
SOCKET_NETWORK socket2793 80
driver88a
SOCKET_NETWORK socket2794 80
driver88b
SOCKET_NETWORK socket2795 80
driver88c
SOCKET_NETWORK socket2796 80
driver88d
SOCKET_NETWORK socket2797 80
driver88a
SOCKET_NETWORK socket2798 80
driver88b
SOCKET_NETWORK socket2799 80
driver88c
SOCKET_NETWORK socket2800 80
driver88d
SOCKET_NETWORK socket2801 80
driver88a
SOCKET_NETWORK socket2802 80
driver88b
SOCKET_NETWORK socket2803 80
driver88c
SOCKET_NETWORK socket2804 80
driver88d
SOCKET_NETWORK socket2805 80
driver88a
SOCKET_NETWORK socket2806 80
driver88b
SOCKET_NETWORK socket2807 80
driver88c
SOCKET_NETWORK socket2808 80
driver88d
SOCKET_NETWORK socket2809 80
driver88a
SOCKET_NETWORK socket2810 80
driver88b
SOCKET_NETWORK socket2811 80
driver88c
SOCKET_NETWORK socket2812 80
driver88d
SOCKET_NETWORK socket2813 80
driver88a
SOCKET_NETWORK socket2814 80
driver88b
SOCKET_NETWORK socket2815 80
driver88c
SOCKET_NETWORK socket2816 80
driver88d
#elif MASTER_NUM23
SOCKET_NETWORK socket2817 80
driver89a
SOCKET_NETWORK socket2818 80
driver89b
SOCKET_NETWORK socket2819 80
driver89c
SOCKET_NETWORK socket2820 80
driver89d
SOCKET_NETWORK socket2821 80
driver89a
SOCKET_NETWORK socket2822 80
driver89b
SOCKET_NETWORK socket2823 80
driver89c
SOCKET_NETWORK socket2824 80
driver89d
SOCKET_NETWORK socket2825 80
driver89a
SOCKET_NETWORK socket2826 80
driver89b
SOCKET_NETWORK socket2827 80
driver89c
SOCKET_NETWORK socket2828 80
driver89d
SOCKET_NETWORK socket2829 80
driver89a
SOCKET_NETWORK socket2830 80
driver89b
SOCKET_NETWORK socket2831 80
driver89c
SOCKET_NETWORK socket2832 80
driver89d
SOCKET_NETWORK socket2833 80
driver89a
SOCKET_NETWORK socket2834 80
driver89b
SOCKET_NETWORK socket2835 80
driver89c
SOCKET_NETWORK socket2836 80
driver89d
SOCKET_NETWORK socket2837 80
driver89a
SOCKET_NETWORK socket2838 80
driver89b
SOCKET_NETWORK socket2839 80
driver89c
SOCKET_NETWORK socket2840 80
driver89d
SOCKET_NETWORK socket2841 80
driver89a
SOCKET_NETWORK socket2842 80
driver89b
SOCKET_NETWORK socket2843 80
driver89c
SOCKET_NETWORK socket2844 80
driver89d
SOCKET_NETWORK socket2845 80
driver89a
SOCKET_NETWORK socket2846 80
driver89b
SOCKET_NETWORK socket2847 80
driver89c
SOCKET_NETWORK socket2848 80
driver89d
SOCKET_NETWORK socket2849 80
driver90a
SOCKET_NETWORK socket2850 80
driver90b
SOCKET_NETWORK socket2851 80
driver90c
SOCKET_NETWORK socket2852 80
driver90d
SOCKET_NETWORK socket2853 80
driver90a
SOCKET_NETWORK socket2854 80
driver90b
SOCKET_NETWORK socket2855 80
driver90c
SOCKET_NETWORK socket2856 80
driver90d
SOCKET_NETWORK socket2857 80
driver90a
SOCKET_NETWORK socket2858 80
driver90b
SOCKET_NETWORK socket2859 80
driver90c
SOCKET_NETWORK socket2860 80
driver90d
SOCKET_NETWORK socket2861 80
driver90a
SOCKET_NETWORK socket2862 80
driver90b
SOCKET_NETWORK socket2863 80
driver90c
SOCKET_NETWORK socket2864 80
driver90d
SOCKET_NETWORK socket2865 80
driver90a
SOCKET_NETWORK socket2866 80
driver90b
SOCKET_NETWORK socket2867 80
driver90c
SOCKET_NETWORK socket2868 80
driver90d
SOCKET_NETWORK socket2869 80
driver90a
SOCKET_NETWORK socket2870 80
driver90b
SOCKET_NETWORK socket2871 80
driver90c
SOCKET_NETWORK socket2872 80
driver90d
SOCKET_NETWORK socket2873 80
driver90a
SOCKET_NETWORK socket2874 80
driver90b
SOCKET_NETWORK socket2875 80
driver90c
SOCKET_NETWORK socket2876 80
driver90d
SOCKET_NETWORK socket2877 80
driver90a
SOCKET_NETWORK socket2878 80
driver90b
SOCKET_NETWORK socket2879 80
driver90c
SOCKET_NETWORK socket2880 80
driver90d
SOCKET_NETWORK socket2881 80
driver91a
SOCKET_NETWORK socket2882 80
driver91b
SOCKET_NETWORK socket2883 80
driver91c
SOCKET_NETWORK socket2884 80
driver91d
SOCKET_NETWORK socket2885 80
driver91a
SOCKET_NETWORK socket2886 80
driver91b
SOCKET_NETWORK socket2887 80
driver91c
SOCKET_NETWORK socket2888 80
driver91d
SOCKET_NETWORK socket2889 80
driver91a
SOCKET_NETWORK socket2890 80
driver91b
SOCKET_NETWORK socket2891 80
driver91c
SOCKET_NETWORK socket2892 80
driver91d
```

```
SOCKET_NETWORK socket2893  80
driver91a
SOCKET_NETWORK socket2894  80
driver91b
SOCKET_NETWORK socket2895  80
driver91c
SOCKET_NETWORK socket2896  80
driver91d
SOCKET_NETWORK socket2897  80
driver91a
SOCKET_NETWORK socket2898  80
driver91b
SOCKET_NETWORK socket2899  80
driver91c
SOCKET_NETWORK socket2900  80
driver91d
SOCKET_NETWORK socket2901  80
driver91a
SOCKET_NETWORK socket2902  80
driver91b
SOCKET_NETWORK socket2903  80
driver91c
SOCKET_NETWORK socket2904  80
driver91d
SOCKET_NETWORK socket2905  80
driver91a
SOCKET_NETWORK socket2906  80
driver91b
SOCKET_NETWORK socket2907  80
driver91c
SOCKET_NETWORK socket2908  80
driver91d
SOCKET_NETWORK socket2909  80
driver91a
SOCKET_NETWORK socket2910  80
driver91b
SOCKET_NETWORK socket2911  80
driver91c
SOCKET_NETWORK socket2912  80
driver91d
SOCKET_NETWORK socket2913  80
driver92a
SOCKET_NETWORK socket2914  80
driver92b
SOCKET_NETWORK socket2915  80
driver92c
SOCKET_NETWORK socket2916  80
driver92d
SOCKET_NETWORK socket2917  80
driver92a
SOCKET_NETWORK socket2918  80
driver92b
SOCKET_NETWORK socket2919  80
driver92c
SOCKET_NETWORK socket2920  80
driver92d
SOCKET_NETWORK socket2921  80
driver92a
SOCKET_NETWORK socket2922  80
driver92b
SOCKET_NETWORK socket2923  80
driver92c
SOCKET_NETWORK socket2924  80
driver92d
SOCKET_NETWORK socket2925  80
driver92a
SOCKET_NETWORK socket2926  80
driver92b
SOCKET_NETWORK socket2927  80
driver92c
SOCKET_NETWORK socket2928  80
driver92d
SOCKET_NETWORK socket2929  80
driver92a
SOCKET_NETWORK socket2930  80
driver92b
SOCKET_NETWORK socket2931  80
driver92c
SOCKET_NETWORK socket2932  80
driver92d
SOCKET_NETWORK socket2933  80
driver92a
SOCKET_NETWORK socket2934  80
driver92b
SOCKET_NETWORK socket2935  80
driver92c
SOCKET_NETWORK socket2936  80
driver92d
SOCKET_NETWORK socket2937  80
driver92a
SOCKET_NETWORK socket2938  80
driver92b
SOCKET_NETWORK socket2939  80
driver92c
SOCKET_NETWORK socket2940  80
driver92d
SOCKET_NETWORK socket2941  80
driver92a
SOCKET_NETWORK socket2942  80
driver92b
SOCKET_NETWORK socket2943  80
driver92c
SOCKET_NETWORK socket2944  80
driver92d
#elif MASTER_NUM24
SOCKET_NETWORK socket2945  80
driver93a
SOCKET_NETWORK socket2946  80
driver93b
SOCKET_NETWORK socket2947  80
driver93c
SOCKET_NETWORK socket2948  80
driver93d
SOCKET_NETWORK socket2949  80
driver93a
SOCKET_NETWORK socket2950  80
driver93b
SOCKET_NETWORK socket2951  80
driver93c
SOCKET_NETWORK socket2952  80
driver93d
SOCKET_NETWORK socket2953  80
driver93a
SOCKET_NETWORK socket2954  80
driver93b
SOCKET_NETWORK socket2955  80
driver93c
SOCKET_NETWORK socket2956  80
driver93d
SOCKET_NETWORK socket2957  80
driver93a
SOCKET_NETWORK socket2958  80
driver93b
SOCKET_NETWORK socket2959  80
driver93c
SOCKET_NETWORK socket2960  80
driver93d
SOCKET_NETWORK socket2961  80
driver93a
SOCKET_NETWORK socket2962  80
driver93b
SOCKET_NETWORK socket2963  80
driver93c
SOCKET_NETWORK socket2964  80
driver93d
SOCKET_NETWORK socket2965  80
driver93a
SOCKET_NETWORK socket2966  80
driver93b
SOCKET_NETWORK socket2967  80
driver93c
SOCKET_NETWORK socket2968  80
driver93d
SOCKET_NETWORK socket2969  80
driver93a
SOCKET_NETWORK socket2970  80
driver93b
SOCKET_NETWORK socket2971  80
driver93c
SOCKET_NETWORK socket2972  80
driver93d
SOCKET_NETWORK socket2973  80
driver93a
SOCKET_NETWORK socket2974  80
driver93b
SOCKET_NETWORK socket2975  80
driver93c
SOCKET_NETWORK socket2976  80
driver93d
SOCKET_NETWORK socket2977  80
driver94a
SOCKET_NETWORK socket2978  80
driver94b
SOCKET_NETWORK socket2979  80
driver94c
SOCKET_NETWORK socket2980  80
driver94d
SOCKET_NETWORK socket2981  80
driver94a
SOCKET_NETWORK socket2982  80
driver94b
SOCKET_NETWORK socket2983  80
driver94c
SOCKET_NETWORK socket2984  80
driver94d
SOCKET_NETWORK socket2985  80
driver94a
SOCKET_NETWORK socket2986  80
driver94b
SOCKET_NETWORK socket2987  80
driver94c
SOCKET_NETWORK socket2988  80
driver94d
SOCKET_NETWORK socket2989  80
driver94a
SOCKET_NETWORK socket2990  80
driver94b
SOCKET_NETWORK socket2991  80
driver94c
SOCKET_NETWORK socket2992  80
driver94d
SOCKET_NETWORK socket2993  80
driver94a
SOCKET_NETWORK socket2994  80
driver94b
SOCKET_NETWORK socket2995  80
driver94c
SOCKET_NETWORK socket2996  80
driver94d
SOCKET_NETWORK socket2997  80
driver94a
SOCKET_NETWORK socket2998  80
driver94b
SOCKET_NETWORK socket2999  80
driver94c
SOCKET_NETWORK socket3000  80
driver94d
SOCKET_NETWORK socket3001  80
driver94a
SOCKET_NETWORK socket3002  80
driver94b
SOCKET_NETWORK socket3003  80
driver94c
SOCKET_NETWORK socket3004  80
driver94d
SOCKET_NETWORK socket3005  80
driver94a
SOCKET_NETWORK socket3006  80
driver94b
SOCKET_NETWORK socket3007  80
driver94c
SOCKET_NETWORK socket3008  80
driver94d
SOCKET_NETWORK socket3009  80
driver95a
SOCKET_NETWORK socket3010  80
driver95b
SOCKET_NETWORK socket3011  80
driver95c
SOCKET_NETWORK socket3012  80
driver95d
SOCKET_NETWORK socket3013  80
driver95a
SOCKET_NETWORK socket3014  80
driver95b
SOCKET_NETWORK socket3015  80
driver95c
```

```
SOCKET_NETWORK socket3016 80
driver95d
SOCKET_NETWORK socket3017 80
driver95a
SOCKET_NETWORK socket3018 80
driver95b
SOCKET_NETWORK socket3019 80
driver95c
SOCKET_NETWORK socket3020 80
driver95d
SOCKET_NETWORK socket3021 80
driver95a
SOCKET_NETWORK socket3022 80
driver95b
SOCKET_NETWORK socket3023 80
driver95c
SOCKET_NETWORK socket3024 80
driver95d
SOCKET_NETWORK socket3025 80
driver95a
SOCKET_NETWORK socket3026 80
driver95b
SOCKET_NETWORK socket3027 80
driver95c
SOCKET_NETWORK socket3028 80
driver95d
SOCKET_NETWORK socket3029 80
driver95a
SOCKET_NETWORK socket3030 80
driver95b
SOCKET_NETWORK socket3031 80
driver95c
SOCKET_NETWORK socket3032 80
driver95d
SOCKET_NETWORK socket3033 80
driver95a
SOCKET_NETWORK socket3034 80
driver95b
SOCKET_NETWORK socket3035 80
driver95c
SOCKET_NETWORK socket3036 80
driver95d
SOCKET_NETWORK socket3037 80
driver95a
SOCKET_NETWORK socket3038 80
driver95b
SOCKET_NETWORK socket3039 80
driver95c
SOCKET_NETWORK socket3040 80
driver95d
SOCKET_NETWORK socket3041 80
driver96a
SOCKET_NETWORK socket3042 80
driver96b
SOCKET_NETWORK socket3043 80
driver96c
SOCKET_NETWORK socket3044 80
driver96d
SOCKET_NETWORK socket3045 80
driver96a
SOCKET_NETWORK socket3046 80
driver96b
SOCKET_NETWORK socket3047 80
driver96c
SOCKET_NETWORK socket3048 80
driver96d
SOCKET_NETWORK socket3049 80
driver96a
SOCKET_NETWORK socket3050 80
driver96b
SOCKET_NETWORK socket3051 80
driver96c
SOCKET_NETWORK socket3052 80
driver96d
SOCKET_NETWORK socket3053 80
driver96a
SOCKET_NETWORK socket3054 80
driver96b
SOCKET_NETWORK socket3055 80
driver96c
SOCKET_NETWORK socket3056 80
driver96d
SOCKET_NETWORK socket3057 80
driver96a
SOCKET_NETWORK socket3058 80
driver96b
SOCKET_NETWORK socket3059 80
driver96c
SOCKET_NETWORK socket3060 80
driver96d
SOCKET_NETWORK socket3061 80
driver96a
SOCKET_NETWORK socket3062 80
driver96b
SOCKET_NETWORK socket3063 80
driver96c
SOCKET_NETWORK socket3064 80
driver96d
SOCKET_NETWORK socket3065 80
driver96a
SOCKET_NETWORK socket3066 80
driver96b
SOCKET_NETWORK socket3067 80
driver96c
SOCKET_NETWORK socket3068 80
driver96d
SOCKET_NETWORK socket3069 80
driver96a
SOCKET_NETWORK socket3070 80
driver96b
SOCKET_NETWORK socket3071 80
driver96c
SOCKET_NETWORK socket3072 80
driver96d
#elif MASTER_NUM25
SOCKET_NETWORK socket3073 80
driver97a
SOCKET_NETWORK socket3074 80
driver97b
SOCKET_NETWORK socket3075 80
driver97c
SOCKET_NETWORK socket3076 80
driver97d
SOCKET_NETWORK socket3077 80
driver97a
SOCKET_NETWORK socket3078 80
driver97b
SOCKET_NETWORK socket3079 80
driver97c
SOCKET_NETWORK socket3080 80
driver97d
SOCKET_NETWORK socket3081 80
driver97a
SOCKET_NETWORK socket3082 80
driver97b
SOCKET_NETWORK socket3083 80
driver97c
SOCKET_NETWORK socket3084 80
driver97d
SOCKET_NETWORK socket3085 80
driver97a
SOCKET_NETWORK socket3086 80
driver97b
SOCKET_NETWORK socket3087 80
driver97c
SOCKET_NETWORK socket3088 80
driver97d
SOCKET_NETWORK socket3089 80
driver97a
SOCKET_NETWORK socket3090 80
driver97b
SOCKET_NETWORK socket3091 80
driver97c
SOCKET_NETWORK socket3092 80
driver97d
SOCKET_NETWORK socket3093 80
driver97a
SOCKET_NETWORK socket3094 80
driver97b
SOCKET_NETWORK socket3095 80
driver97c
SOCKET_NETWORK socket3096 80
driver97d
SOCKET_NETWORK socket3097 80
driver97a
SOCKET_NETWORK socket3098 80
driver97b
SOCKET_NETWORK socket3099 80
driver97c
SOCKET_NETWORK socket3100 80
driver97d
SOCKET_NETWORK socket3101 80
driver97a
SOCKET_NETWORK socket3102 80
driver97b
SOCKET_NETWORK socket3103 80
driver97c
SOCKET_NETWORK socket3104 80
driver97d
SOCKET_NETWORK socket3105 80
driver98a
SOCKET_NETWORK socket3106 80
driver98b
SOCKET_NETWORK socket3107 80
driver98c
SOCKET_NETWORK socket3108 80
driver98d
SOCKET_NETWORK socket3109 80
driver98a
SOCKET_NETWORK socket3110 80
driver98b
SOCKET_NETWORK socket3111 80
driver98c
SOCKET_NETWORK socket3112 80
driver98d
SOCKET_NETWORK socket3113 80
driver98a
SOCKET_NETWORK socket3114 80
driver98b
SOCKET_NETWORK socket3115 80
driver98c
SOCKET_NETWORK socket3116 80
driver98d
SOCKET_NETWORK socket3117 80
driver98a
SOCKET_NETWORK socket3118 80
driver98b
SOCKET_NETWORK socket3119 80
driver98c
SOCKET_NETWORK socket3120 80
driver98d
SOCKET_NETWORK socket3121 80
driver98a
SOCKET_NETWORK socket3122 80
driver98b
SOCKET_NETWORK socket3123 80
driver98c
SOCKET_NETWORK socket3124 80
driver98d
SOCKET_NETWORK socket3125 80
driver98a
SOCKET_NETWORK socket3126 80
driver98b
SOCKET_NETWORK socket3127 80
driver98c
SOCKET_NETWORK socket3128 80
driver98d
SOCKET_NETWORK socket3129 80
driver98a
SOCKET_NETWORK socket3130 80
driver98b
SOCKET_NETWORK socket3131 80
driver98c
SOCKET_NETWORK socket3132 80
driver98d
SOCKET_NETWORK socket3133 80
driver98a
SOCKET_NETWORK socket3134 80
driver98b
SOCKET_NETWORK socket3135 80
driver98c
SOCKET_NETWORK socket3136 80
driver98d
SOCKET_NETWORK socket3137 80
driver99a
SOCKET_NETWORK socket3138 80
driver99b
```

```
SOCKET_NETWORK socket3139 80
driver99c
SOCKET_NETWORK socket3140 80
driver99d
SOCKET_NETWORK socket3141 80
driver99a
SOCKET_NETWORK socket3142 80
driver99b
SOCKET_NETWORK socket3143 80
driver99c
SOCKET_NETWORK socket3144 80
driver99d
SOCKET_NETWORK socket3145 80
driver99a
SOCKET_NETWORK socket3146 80
driver99b
SOCKET_NETWORK socket3147 80
driver99c
SOCKET_NETWORK socket3148 80
driver99d
SOCKET_NETWORK socket3149 80
driver99a
SOCKET_NETWORK socket3150 80
driver99b
SOCKET_NETWORK socket3151 80
driver99c
SOCKET_NETWORK socket3152 80
driver99d
SOCKET_NETWORK socket3153 80
driver99a
SOCKET_NETWORK socket3154 80
driver99b
SOCKET_NETWORK socket3155 80
driver99c
SOCKET_NETWORK socket3156 80
driver99d
SOCKET_NETWORK socket3157 80
driver99a
SOCKET_NETWORK socket3158 80
driver99b
SOCKET_NETWORK socket3159 80
driver99c
SOCKET_NETWORK socket3160 80
driver99d
SOCKET_NETWORK socket3161 80
driver99a
SOCKET_NETWORK socket3162 80
driver99b
SOCKET_NETWORK socket3163 80
driver99c
SOCKET_NETWORK socket3164 80
driver99d
SOCKET_NETWORK socket3165 80
driver99a
SOCKET_NETWORK socket3166 80
driver99b
SOCKET_NETWORK socket3167 80
driver99c
SOCKET_NETWORK socket3168 80
driver99d
SOCKET_NETWORK socket3169 80
driver100a
SOCKET_NETWORK socket3170 80
driver100b
SOCKET_NETWORK socket3171 80
driver100c
SOCKET_NETWORK socket3172 80
driver100d
SOCKET_NETWORK socket3173 80
driver100a
SOCKET_NETWORK socket3174 80
driver100b
SOCKET_NETWORK socket3175 80
driver100c
SOCKET_NETWORK socket3176 80
driver100d
SOCKET_NETWORK socket3177 80
driver100a
SOCKET_NETWORK socket3178 80
driver100b
SOCKET_NETWORK socket3179 80
driver100c
SOCKET_NETWORK socket3180 80
driver100d
SOCKET_NETWORK socket3181 80
driver100a
SOCKET_NETWORK socket3182 80
driver100b
SOCKET_NETWORK socket3183 80
driver100c
SOCKET_NETWORK socket3184 80
driver100d
SOCKET_NETWORK socket3185 80
driver100a
SOCKET_NETWORK socket3186 80
driver100b
SOCKET_NETWORK socket3187 80
driver100c
SOCKET_NETWORK socket3188 80
driver100d
SOCKET_NETWORK socket3189 80
driver100a
SOCKET_NETWORK socket3190 80
driver100b
SOCKET_NETWORK socket3191 80
driver100c
SOCKET_NETWORK socket3192 80
driver100d
SOCKET_NETWORK socket3193 80
driver100a
SOCKET_NETWORK socket3194 80
driver100b
SOCKET_NETWORK socket3195 80
driver100c
SOCKET_NETWORK socket3196 80
driver100d
SOCKET_NETWORK socket3197 80
driver100a
SOCKET_NETWORK socket3198 80
driver100b
SOCKET_NETWORK socket3199 80
driver100c
SOCKET_NETWORK socket3200 80
driver100d
#elif MASTER_NUM26
SOCKET_NETWORK socket3201 80
driver101a
SOCKET_NETWORK socket3202 80
driver101b
SOCKET_NETWORK socket3203 80
driver101c
SOCKET_NETWORK socket3204 80
driver101d
SOCKET_NETWORK socket3205 80
driver101a
SOCKET_NETWORK socket3206 80
driver101b
SOCKET_NETWORK socket3207 80
driver101c
SOCKET_NETWORK socket3208 80
driver101d
SOCKET_NETWORK socket3209 80
driver101a
SOCKET_NETWORK socket3210 80
driver101b
SOCKET_NETWORK socket3211 80
driver101c
SOCKET_NETWORK socket3212 80
driver101d
SOCKET_NETWORK socket3213 80
driver101a
SOCKET_NETWORK socket3214 80
driver101b
SOCKET_NETWORK socket3215 80
driver101c
SOCKET_NETWORK socket3216 80
driver101d
SOCKET_NETWORK socket3217 80
driver101a
SOCKET_NETWORK socket3218 80
driver101b
SOCKET_NETWORK socket3219 80
driver101c
SOCKET_NETWORK socket3220 80
driver101d
SOCKET_NETWORK socket3221 80
driver101a
SOCKET_NETWORK socket3222 80
driver101b
SOCKET_NETWORK socket3223 80
driver101c
SOCKET_NETWORK socket3224 80
driver101d
SOCKET_NETWORK socket3225 80
driver101a
SOCKET_NETWORK socket3226 80
driver101b
SOCKET_NETWORK socket3227 80
driver101c
SOCKET_NETWORK socket3228 80
driver101d
SOCKET_NETWORK socket3229 80
driver101a
SOCKET_NETWORK socket3230 80
driver101b
SOCKET_NETWORK socket3231 80
driver101c
SOCKET_NETWORK socket3232 80
driver101d
SOCKET_NETWORK socket3233 80
driver102a
SOCKET_NETWORK socket3234 80
driver102b
SOCKET_NETWORK socket3235 80
driver102c
SOCKET_NETWORK socket3236 80
driver102d
SOCKET_NETWORK socket3237 80
driver102a
SOCKET_NETWORK socket3238 80
driver102b
SOCKET_NETWORK socket3239 80
driver102c
SOCKET_NETWORK socket3240 80
driver102d
SOCKET_NETWORK socket3241 80
driver102a
SOCKET_NETWORK socket3242 80
driver102b
SOCKET_NETWORK socket3243 80
driver102c
SOCKET_NETWORK socket3244 80
driver102d
SOCKET_NETWORK socket3245 80
driver102a
SOCKET_NETWORK socket3246 80
driver102b
SOCKET_NETWORK socket3247 80
driver102c
SOCKET_NETWORK socket3248 80
driver102d
SOCKET_NETWORK socket3249 80
driver102a
SOCKET_NETWORK socket3250 80
driver102b
SOCKET_NETWORK socket3251 80
driver102c
SOCKET_NETWORK socket3252 80
driver102d
SOCKET_NETWORK socket3253 80
driver102a
SOCKET_NETWORK socket3254 80
driver102b
SOCKET_NETWORK socket3255 80
driver102c
SOCKET_NETWORK socket3256 80
driver102d
SOCKET_NETWORK socket3257 80
driver102a
SOCKET_NETWORK socket3258 80
driver102b
SOCKET_NETWORK socket3259 80
driver102c
SOCKET_NETWORK socket3260 80
driver102d
SOCKET_NETWORK socket3261 80
driver102a
```

```
SOCKET_NETWORK socket3262 80
driver102b
SOCKET_NETWORK socket3263 80
driver102c
SOCKET_NETWORK socket3264 80
driver102d
SOCKET_NETWORK socket3265 80
driver103a
SOCKET_NETWORK socket3266 80
driver103b
SOCKET_NETWORK socket3267 80
driver103c
SOCKET_NETWORK socket3268 80
driver103d
SOCKET_NETWORK socket3269 80
driver103a
SOCKET_NETWORK socket3270 80
driver103b
SOCKET_NETWORK socket3271 80
driver103c
SOCKET_NETWORK socket3272 80
driver103d
SOCKET_NETWORK socket3273 80
driver103a
SOCKET_NETWORK socket3274 80
driver103b
SOCKET_NETWORK socket3275 80
driver103c
SOCKET_NETWORK socket3276 80
driver103d
SOCKET_NETWORK socket3277 80
driver103a
SOCKET_NETWORK socket3278 80
driver103b
SOCKET_NETWORK socket3279 80
driver103c
SOCKET_NETWORK socket3280 80
driver103d
SOCKET_NETWORK socket3281 80
driver103a
SOCKET_NETWORK socket3282 80
driver103b
SOCKET_NETWORK socket3283 80
driver103c
SOCKET_NETWORK socket3284 80
driver103d
SOCKET_NETWORK socket3285 80
driver103a
SOCKET_NETWORK socket3286 80
driver103b
SOCKET_NETWORK socket3287 80
driver103c
SOCKET_NETWORK socket3288 80
driver103d
SOCKET_NETWORK socket3289 80
driver103a
SOCKET_NETWORK socket3290 80
driver103b
SOCKET_NETWORK socket3291 80
driver103c
SOCKET_NETWORK socket3292 80
driver103d
SOCKET_NETWORK socket3293 80
driver103a
SOCKET_NETWORK socket3294 80
driver103b
SOCKET_NETWORK socket3295 80
driver103c
SOCKET_NETWORK socket3296 80
driver103d
SOCKET_NETWORK socket3297 80
driver104a
SOCKET_NETWORK socket3298 80
driver104b
SOCKET_NETWORK socket3299 80
driver104c
SOCKET_NETWORK socket3300 80
driver104d
SOCKET_NETWORK socket3301 80
driver104a
SOCKET_NETWORK socket3302 80
driver104b
SOCKET_NETWORK socket3303 80
driver104c
SOCKET_NETWORK socket3304 80
driver104d
SOCKET_NETWORK socket3305 80
driver104a
SOCKET_NETWORK socket3306 80
driver104b
SOCKET_NETWORK socket3307 80
driver104c
SOCKET_NETWORK socket3308 80
driver104d
SOCKET_NETWORK socket3309 80
driver104a
SOCKET_NETWORK socket3310 80
driver104b
SOCKET_NETWORK socket3311 80
driver104c
SOCKET_NETWORK socket3312 80
driver104d
SOCKET_NETWORK socket3313 80
driver104a
SOCKET_NETWORK socket3314 80
driver104b
SOCKET_NETWORK socket3315 80
driver104c
SOCKET_NETWORK socket3316 80
driver104d
SOCKET_NETWORK socket3317 80
driver104a
SOCKET_NETWORK socket3318 80
driver104b
SOCKET_NETWORK socket3319 80
driver104c
SOCKET_NETWORK socket3320 80
driver104d
SOCKET_NETWORK socket3321 80
driver104a
SOCKET_NETWORK socket3322 80
driver104b
SOCKET_NETWORK socket3323 80
driver104c
SOCKET_NETWORK socket3324 80
driver104d
SOCKET_NETWORK socket3325 80
driver104a
SOCKET_NETWORK socket3326 80
driver104b
SOCKET_NETWORK socket3327 80
driver104c
SOCKET_NETWORK socket3328 80
driver104d
#elif MASTER_NUM27
SOCKET_NETWORK socket3329 80
driver105a
SOCKET_NETWORK socket3330 80
driver105b
SOCKET_NETWORK socket3331 80
driver105c
SOCKET_NETWORK socket3332 80
driver105d
SOCKET_NETWORK socket3333 80
driver105a
SOCKET_NETWORK socket3334 80
driver105b
SOCKET_NETWORK socket3335 80
driver105c
SOCKET_NETWORK socket3336 80
driver105d
SOCKET_NETWORK socket3337 80
driver105a
SOCKET_NETWORK socket3338 80
driver105b
SOCKET_NETWORK socket3339 80
driver105c
SOCKET_NETWORK socket3340 80
driver105d
SOCKET_NETWORK socket3341 80
driver105a
SOCKET_NETWORK socket3342 80
driver105b
SOCKET_NETWORK socket3343 80
driver105c
SOCKET_NETWORK socket3344 80
driver105d
SOCKET_NETWORK socket3345 80
driver105a
SOCKET_NETWORK socket3346 80
driver105b
SOCKET_NETWORK socket3347 80
driver105c
SOCKET_NETWORK socket3348 80
driver105d
SOCKET_NETWORK socket3349 80
driver105a
SOCKET_NETWORK socket3350 80
driver105b
SOCKET_NETWORK socket3351 80
driver105c
SOCKET_NETWORK socket3352 80
driver105d
SOCKET_NETWORK socket3353 80
driver105a
SOCKET_NETWORK socket3354 80
driver105b
SOCKET_NETWORK socket3355 80
driver105c
SOCKET_NETWORK socket3356 80
driver105d
SOCKET_NETWORK socket3357 80
driver105a
SOCKET_NETWORK socket3358 80
driver105b
SOCKET_NETWORK socket3359 80
driver105c
SOCKET_NETWORK socket3360 80
driver105d
SOCKET_NETWORK socket3361 80
driver106a
SOCKET_NETWORK socket3362 80
driver106b
SOCKET_NETWORK socket3363 80
driver106c
SOCKET_NETWORK socket3364 80
driver106d
SOCKET_NETWORK socket3365 80
driver106a
SOCKET_NETWORK socket3366 80
driver106b
SOCKET_NETWORK socket3367 80
driver106c
SOCKET_NETWORK socket3368 80
driver106d
SOCKET_NETWORK socket3369 80
driver106a
SOCKET_NETWORK socket3370 80
driver106b
SOCKET_NETWORK socket3371 80
driver106c
SOCKET_NETWORK socket3372 80
driver106d
SOCKET_NETWORK socket3373 80
driver106a
SOCKET_NETWORK socket3374 80
driver106b
SOCKET_NETWORK socket3375 80
driver106c
SOCKET_NETWORK socket3376 80
driver106d
SOCKET_NETWORK socket3377 80
driver106a
SOCKET_NETWORK socket3378 80
driver106b
SOCKET_NETWORK socket3379 80
driver106c
SOCKET_NETWORK socket3380 80
driver106d
SOCKET_NETWORK socket3381 80
driver106a
SOCKET_NETWORK socket3382 80
driver106b
SOCKET_NETWORK socket3383 80
driver106c
SOCKET_NETWORK socket3384 80
driver106d
```

```
SOCKET_NETWORK socket3385  80
driver106a
SOCKET_NETWORK socket3386  80
driver106b
SOCKET_NETWORK socket3387  80
driver106c
SOCKET_NETWORK socket3388  80
driver106d
SOCKET_NETWORK socket3389  80
driver106a
SOCKET_NETWORK socket3390  80
driver106b
SOCKET_NETWORK socket3391  80
driver106c
SOCKET_NETWORK socket3392  80
driver106d
SOCKET_NETWORK socket3393  80
driver107a
SOCKET_NETWORK socket3394  80
driver107b
SOCKET_NETWORK socket3395  80
driver107c
SOCKET_NETWORK socket3396  80
driver107d
SOCKET_NETWORK socket3397  80
driver107a
SOCKET_NETWORK socket3398  80
driver107b
SOCKET_NETWORK socket3399  80
driver107c
SOCKET_NETWORK socket3400  80
driver107d
SOCKET_NETWORK socket3401  80
driver107a
SOCKET_NETWORK socket3402  80
driver107b
SOCKET_NETWORK socket3403  80
driver107c
SOCKET_NETWORK socket3404  80
driver107d
SOCKET_NETWORK socket3405  80
driver107a
SOCKET_NETWORK socket3406  80
driver107b
SOCKET_NETWORK socket3407  80
driver107c
SOCKET_NETWORK socket3408  80
driver107d
SOCKET_NETWORK socket3409  80
driver107a
SOCKET_NETWORK socket3410  80
driver107b
SOCKET_NETWORK socket3411  80
driver107c
SOCKET_NETWORK socket3412  80
driver107d
SOCKET_NETWORK socket3413  80
driver107a
SOCKET_NETWORK socket3414  80
driver107b
SOCKET_NETWORK socket3415  80
driver107c
SOCKET_NETWORK socket3416  80
driver107d
SOCKET_NETWORK socket3417  80
driver107a
SOCKET_NETWORK socket3418  80
driver107b
SOCKET_NETWORK socket3419  80
driver107c
SOCKET_NETWORK socket3420  80
driver107d
SOCKET_NETWORK socket3421  80
driver107a
SOCKET_NETWORK socket3422  80
driver107b
SOCKET_NETWORK socket3423  80
driver107c
SOCKET_NETWORK socket3424  80
driver107d
SOCKET_NETWORK socket3425  80
driver108a
SOCKET_NETWORK socket3426  80
driver108b
SOCKET_NETWORK socket3427  80
driver108c
SOCKET_NETWORK socket3428  80
driver108d
SOCKET_NETWORK socket3429  80
driver108a
SOCKET_NETWORK socket3430  80
driver108b
SOCKET_NETWORK socket3431  80
driver108c
SOCKET_NETWORK socket3432  80
driver108d
SOCKET_NETWORK socket3433  80
driver108a
SOCKET_NETWORK socket3434  80
driver108b
SOCKET_NETWORK socket3435  80
driver108c
SOCKET_NETWORK socket3436  80
driver108d
SOCKET_NETWORK socket3437  80
driver108a
SOCKET_NETWORK socket3438  80
driver108b
SOCKET_NETWORK socket3439  80
driver108c
SOCKET_NETWORK socket3440  80
driver108d
SOCKET_NETWORK socket3441  80
driver108a
SOCKET_NETWORK socket3442  80
driver108b
SOCKET_NETWORK socket3443  80
driver108c
SOCKET_NETWORK socket3444  80
driver108d
SOCKET_NETWORK socket3445  80
driver108a
SOCKET_NETWORK socket3446  80
driver108b
SOCKET_NETWORK socket3447  80
driver108c
SOCKET_NETWORK socket3448  80
driver108d
SOCKET_NETWORK socket3449  80
driver108a
SOCKET_NETWORK socket3450  80
driver108b
SOCKET_NETWORK socket3451  80
driver108c
SOCKET_NETWORK socket3452  80
driver108d
SOCKET_NETWORK socket3453  80
driver108a
SOCKET_NETWORK socket3454  80
driver108b
SOCKET_NETWORK socket3455  80
driver108c
SOCKET_NETWORK socket3456  80
driver108d
#elif MASTER_NUM28
SOCKET_NETWORK socket3457  80
driver109a
SOCKET_NETWORK socket3458  80
driver109b
SOCKET_NETWORK socket3459  80
driver109c
SOCKET_NETWORK socket3460  80
driver109d
SOCKET_NETWORK socket3461  80
driver109a
SOCKET_NETWORK socket3462  80
driver109b
SOCKET_NETWORK socket3463  80
driver109c
SOCKET_NETWORK socket3464  80
driver109d
SOCKET_NETWORK socket3465  80
driver109a
SOCKET_NETWORK socket3466  80
driver109b
SOCKET_NETWORK socket3467  80
driver109c
SOCKET_NETWORK socket3468  80
driver109d
SOCKET_NETWORK socket3469  80
driver109a
SOCKET_NETWORK socket3470  80
driver109b
SOCKET_NETWORK socket3471  80
driver109c
SOCKET_NETWORK socket3472  80
driver109d
SOCKET_NETWORK socket3473  80
driver109a
SOCKET_NETWORK socket3474  80
driver109b
SOCKET_NETWORK socket3475  80
driver109c
SOCKET_NETWORK socket3476  80
driver109d
SOCKET_NETWORK socket3477  80
driver109a
SOCKET_NETWORK socket3478  80
driver109b
SOCKET_NETWORK socket3479  80
driver109c
SOCKET_NETWORK socket3480  80
driver109d
SOCKET_NETWORK socket3481  80
driver109a
SOCKET_NETWORK socket3482  80
driver109b
SOCKET_NETWORK socket3483  80
driver109c
SOCKET_NETWORK socket3484  80
driver109d
SOCKET_NETWORK socket3485  80
driver109a
SOCKET_NETWORK socket3486  80
driver109b
SOCKET_NETWORK socket3487  80
driver109c
SOCKET_NETWORK socket3488  80
driver109d
SOCKET_NETWORK socket3489  80
driver110a
SOCKET_NETWORK socket3490  80
driver110b
SOCKET_NETWORK socket3491  80
driver110c
SOCKET_NETWORK socket3492  80
driver110d
SOCKET_NETWORK socket3493  80
driver110a
SOCKET_NETWORK socket3494  80
driver110b
SOCKET_NETWORK socket3495  80
driver110c
SOCKET_NETWORK socket3496  80
driver110d
SOCKET_NETWORK socket3497  80
driver110a
SOCKET_NETWORK socket3498  80
driver110b
SOCKET_NETWORK socket3499  80
driver110c
SOCKET_NETWORK socket3500  80
driver110d
SOCKET_NETWORK socket3501  80
driver110a
SOCKET_NETWORK socket3502  80
driver110b
SOCKET_NETWORK socket3503  80
driver110c
SOCKET_NETWORK socket3504  80
driver110d
SOCKET_NETWORK socket3505  80
driver110a
SOCKET_NETWORK socket3506  80
driver110b
SOCKET_NETWORK socket3507  80
driver110c
```

```
SOCKET_NETWORK socket3508 80
driver110d
SOCKET_NETWORK socket3509 80
driver110a
SOCKET_NETWORK socket3510 80
driver110b
SOCKET_NETWORK socket3511 80
driver110c
SOCKET_NETWORK socket3512 80
driver110d
SOCKET_NETWORK socket3513 80
driver110a
SOCKET_NETWORK socket3514 80
driver110b
SOCKET_NETWORK socket3515 80
driver110c
SOCKET_NETWORK socket3516 80
driver110d
SOCKET_NETWORK socket3517 80
driver110a
SOCKET_NETWORK socket3518 80
driver110b
SOCKET_NETWORK socket3519 80
driver110c
SOCKET_NETWORK socket3520 80
driver110d
SOCKET_NETWORK socket3521 80
driver111a
SOCKET_NETWORK socket3522 80
driver111b
SOCKET_NETWORK socket3523 80
driver111c
SOCKET_NETWORK socket3524 80
driver111d
SOCKET_NETWORK socket3525 80
driver111a
SOCKET_NETWORK socket3526 80
driver111b
SOCKET_NETWORK socket3527 80
driver111c
SOCKET_NETWORK socket3528 80
driver111d
SOCKET_NETWORK socket3529 80
driver111a
SOCKET_NETWORK socket3530 80
driver111b
SOCKET_NETWORK socket3531 80
driver111c
SOCKET_NETWORK socket3532 80
driver111d
SOCKET_NETWORK socket3533 80
driver111a
SOCKET_NETWORK socket3534 80
driver111b
SOCKET_NETWORK socket3535 80
driver111c
SOCKET_NETWORK socket3536 80
driver111d
SOCKET_NETWORK socket3537 80
driver111a
SOCKET_NETWORK socket3538 80
driver111b
SOCKET_NETWORK socket3539 80
driver111c
SOCKET_NETWORK socket3540 80
driver111d
SOCKET_NETWORK socket3541 80
driver111a
SOCKET_NETWORK socket3542 80
driver111b
SOCKET_NETWORK socket3543 80
driver111c
SOCKET_NETWORK socket3544 80
driver111d
SOCKET_NETWORK socket3545 80
driver111a
SOCKET_NETWORK socket3546 80
driver111b
SOCKET_NETWORK socket3547 80
driver111c
SOCKET_NETWORK socket3548 80
driver111d
SOCKET_NETWORK socket3549 80
driver111a
SOCKET_NETWORK socket3550 80
driver111b
SOCKET_NETWORK socket3551 80
driver111c
SOCKET_NETWORK socket3552 80
driver111d
SOCKET_NETWORK socket3553 80
driver112a
SOCKET_NETWORK socket3554 80
driver112b
SOCKET_NETWORK socket3555 80
driver112c
SOCKET_NETWORK socket3556 80
driver112d
SOCKET_NETWORK socket3557 80
driver112a
SOCKET_NETWORK socket3558 80
driver112b
SOCKET_NETWORK socket3559 80
driver112c
SOCKET_NETWORK socket3560 80
driver112d
SOCKET_NETWORK socket3561 80
driver112a
SOCKET_NETWORK socket3562 80
driver112b
SOCKET_NETWORK socket3563 80
driver112c
SOCKET_NETWORK socket3564 80
driver112d
SOCKET_NETWORK socket3565 80
driver112a
SOCKET_NETWORK socket3566 80
driver112b
SOCKET_NETWORK socket3567 80
driver112c
SOCKET_NETWORK socket3568 80
driver112d
SOCKET_NETWORK socket3569 80
driver112a
SOCKET_NETWORK socket3570 80
driver112b
SOCKET_NETWORK socket3571 80
driver112c
SOCKET_NETWORK socket3572 80
driver112d
SOCKET_NETWORK socket3573 80
driver112a
SOCKET_NETWORK socket3574 80
driver112b
SOCKET_NETWORK socket3575 80
driver112c
SOCKET_NETWORK socket3576 80
driver112d
SOCKET_NETWORK socket3577 80
driver112a
SOCKET_NETWORK socket3578 80
driver112b
SOCKET_NETWORK socket3579 80
driver112c
SOCKET_NETWORK socket3580 80
driver112d
SOCKET_NETWORK socket3581 80
driver112a
SOCKET_NETWORK socket3582 80
driver112b
SOCKET_NETWORK socket3583 80
driver112c
SOCKET_NETWORK socket3584 80
driver112d
#elif MASTER_NUM29
SOCKET_NETWORK socket3585 80
driver113a
SOCKET_NETWORK socket3586 80
driver113b
SOCKET_NETWORK socket3587 80
driver113c
SOCKET_NETWORK socket3588 80
driver113d
SOCKET_NETWORK socket3589 80
driver113a
SOCKET_NETWORK socket3590 80
driver113b
SOCKET_NETWORK socket3591 80
driver113c
SOCKET_NETWORK socket3592 80
driver113d
SOCKET_NETWORK socket3593 80
driver113a
SOCKET_NETWORK socket3594 80
driver113b
SOCKET_NETWORK socket3595 80
driver113c
SOCKET_NETWORK socket3596 80
driver113d
SOCKET_NETWORK socket3597 80
driver113a
SOCKET_NETWORK socket3598 80
driver113b
SOCKET_NETWORK socket3599 80
driver113c
SOCKET_NETWORK socket3600 80
driver113d
SOCKET_NETWORK socket3601 80
driver113a
SOCKET_NETWORK socket3602 80
driver113b
SOCKET_NETWORK socket3603 80
driver113c
SOCKET_NETWORK socket3604 80
driver113d
SOCKET_NETWORK socket3605 80
driver113a
SOCKET_NETWORK socket3606 80
driver113b
SOCKET_NETWORK socket3607 80
driver113c
SOCKET_NETWORK socket3608 80
driver113d
SOCKET_NETWORK socket3609 80
driver113a
SOCKET_NETWORK socket3610 80
driver113b
SOCKET_NETWORK socket3611 80
driver113c
SOCKET_NETWORK socket3612 80
driver113d
SOCKET_NETWORK socket3613 80
driver113a
SOCKET_NETWORK socket3614 80
driver113b
SOCKET_NETWORK socket3615 80
driver113c
SOCKET_NETWORK socket3616 80
driver113d
SOCKET_NETWORK socket3617 80
driver114a
SOCKET_NETWORK socket3618 80
driver114b
SOCKET_NETWORK socket3619 80
driver114c
SOCKET_NETWORK socket3620 80
driver114d
SOCKET_NETWORK socket3621 80
driver114a
SOCKET_NETWORK socket3622 80
driver114b
SOCKET_NETWORK socket3623 80
driver114c
SOCKET_NETWORK socket3624 80
driver114d
SOCKET_NETWORK socket3625 80
driver114a
SOCKET_NETWORK socket3626 80
driver114b
SOCKET_NETWORK socket3627 80
driver114c
SOCKET_NETWORK socket3628 80
driver114d
SOCKET_NETWORK socket3629 80
driver114a
SOCKET_NETWORK socket3630 80
driver114b
```

```
SOCKET_NETWORK socket3631  80
driver114c
SOCKET_NETWORK socket3632  80
driver114d
SOCKET_NETWORK socket3633  80
driver114a
SOCKET_NETWORK socket3634  80
driver114b
SOCKET_NETWORK socket3635  80
driver114c
SOCKET_NETWORK socket3636  80
driver114d
SOCKET_NETWORK socket3637  80
driver114a
SOCKET_NETWORK socket3638  80
driver114b
SOCKET_NETWORK socket3639  80
driver114c
SOCKET_NETWORK socket3640  80
driver114d
SOCKET_NETWORK socket3641  80
driver114a
SOCKET_NETWORK socket3642  80
driver114b
SOCKET_NETWORK socket3643  80
driver114c
SOCKET_NETWORK socket3644  80
driver114d
SOCKET_NETWORK socket3645  80
driver114a
SOCKET_NETWORK socket3646  80
driver114b
SOCKET_NETWORK socket3647  80
driver114c
SOCKET_NETWORK socket3648  80
driver114d
SOCKET_NETWORK socket3649  80
driver115a
SOCKET_NETWORK socket3650  80
driver115b
SOCKET_NETWORK socket3651  80
driver115c
SOCKET_NETWORK socket3652  80
driver115d
SOCKET_NETWORK socket3653  80
driver115a
SOCKET_NETWORK socket3654  80
driver115b
SOCKET_NETWORK socket3655  80
driver115c
SOCKET_NETWORK socket3656  80
driver115d
SOCKET_NETWORK socket3657  80
driver115a
SOCKET_NETWORK socket3658  80
driver115b
SOCKET_NETWORK socket3659  80
driver115c
SOCKET_NETWORK socket3660  80
driver115d
SOCKET_NETWORK socket3661  80
driver115a
SOCKET_NETWORK socket3662  80
driver115b
SOCKET_NETWORK socket3663  80
driver115c
SOCKET_NETWORK socket3664  80
driver115d
SOCKET_NETWORK socket3665  80
driver115a
SOCKET_NETWORK socket3666  80
driver115b
SOCKET_NETWORK socket3667  80
driver115c
SOCKET_NETWORK socket3668  80
driver115d
SOCKET_NETWORK socket3669  80
driver115a
SOCKET_NETWORK socket3670  80
driver115b
SOCKET_NETWORK socket3671  80
driver115c
SOCKET_NETWORK socket3672  80
driver115d
SOCKET_NETWORK socket3673  80
driver115a
SOCKET_NETWORK socket3674  80
driver115b
SOCKET_NETWORK socket3675  80
driver115c
SOCKET_NETWORK socket3676  80
driver115d
SOCKET_NETWORK socket3677  80
driver115a
SOCKET_NETWORK socket3678  80
driver115b
SOCKET_NETWORK socket3679  80
driver115c
SOCKET_NETWORK socket3680  80
driver115d
SOCKET_NETWORK socket3681  80
driver116a
SOCKET_NETWORK socket3682  80
driver116b
SOCKET_NETWORK socket3683  80
driver116c
SOCKET_NETWORK socket3684  80
driver116d
SOCKET_NETWORK socket3685  80
driver116a
SOCKET_NETWORK socket3686  80
driver116b
SOCKET_NETWORK socket3687  80
driver116c
SOCKET_NETWORK socket3688  80
driver116d
SOCKET_NETWORK socket3689  80
driver116a
SOCKET_NETWORK socket3690  80
driver116b
SOCKET_NETWORK socket3691  80
driver116c
SOCKET_NETWORK socket3692  80
driver116d
SOCKET_NETWORK socket3693  80
driver116a
SOCKET_NETWORK socket3694  80
driver116b
SOCKET_NETWORK socket3695  80
driver116c
SOCKET_NETWORK socket3696  80
driver116d
SOCKET_NETWORK socket3697  80
driver116a
SOCKET_NETWORK socket3698  80
driver116b
SOCKET_NETWORK socket3699  80
driver116c
SOCKET_NETWORK socket3700  80
driver116d
SOCKET_NETWORK socket3701  80
driver116a
SOCKET_NETWORK socket3702  80
driver116b
SOCKET_NETWORK socket3703  80
driver116c
SOCKET_NETWORK socket3704  80
driver116d
SOCKET_NETWORK socket3705  80
driver116a
SOCKET_NETWORK socket3706  80
driver116b
SOCKET_NETWORK socket3707  80
driver116c
SOCKET_NETWORK socket3708  80
driver116d
SOCKET_NETWORK socket3709  80
driver116a
SOCKET_NETWORK socket3710  80
driver116b
SOCKET_NETWORK socket3711  80
driver116c
SOCKET_NETWORK socket3712  80
driver116d
#elif MASTER_NUM30
SOCKET_NETWORK socket3713  80
driver117a
SOCKET_NETWORK socket3714  80
driver117b
SOCKET_NETWORK socket3715  80
driver117c
SOCKET_NETWORK socket3716  80
driver117d
SOCKET_NETWORK socket3717  80
driver117a
SOCKET_NETWORK socket3718  80
driver117b
SOCKET_NETWORK socket3719  80
driver117c
SOCKET_NETWORK socket3720  80
driver117d
SOCKET_NETWORK socket3721  80
driver117a
SOCKET_NETWORK socket3722  80
driver117b
SOCKET_NETWORK socket3723  80
driver117c
SOCKET_NETWORK socket3724  80
driver117d
SOCKET_NETWORK socket3725  80
driver117a
SOCKET_NETWORK socket3726  80
driver117b
SOCKET_NETWORK socket3727  80
driver117c
SOCKET_NETWORK socket3728  80
driver117d
SOCKET_NETWORK socket3729  80
driver117a
SOCKET_NETWORK socket3730  80
driver117b
SOCKET_NETWORK socket3731  80
driver117c
SOCKET_NETWORK socket3732  80
driver117d
SOCKET_NETWORK socket3733  80
driver117a
SOCKET_NETWORK socket3734  80
driver117b
SOCKET_NETWORK socket3735  80
driver117c
SOCKET_NETWORK socket3736  80
driver117d
SOCKET_NETWORK socket3737  80
driver117a
SOCKET_NETWORK socket3738  80
driver117b
SOCKET_NETWORK socket3739  80
driver117c
SOCKET_NETWORK socket3740  80
driver117d
SOCKET_NETWORK socket3741  80
driver117a
SOCKET_NETWORK socket3742  80
driver117b
SOCKET_NETWORK socket3743  80
driver117c
SOCKET_NETWORK socket3744  80
driver117d
SOCKET_NETWORK socket3745  80
driver118a
SOCKET_NETWORK socket3746  80
driver118b
SOCKET_NETWORK socket3747  80
driver118c
SOCKET_NETWORK socket3748  80
driver118d
SOCKET_NETWORK socket3749  80
driver118a
SOCKET_NETWORK socket3750  80
driver118b
SOCKET_NETWORK socket3751  80
driver118c
SOCKET_NETWORK socket3752  80
driver118d
SOCKET_NETWORK socket3753  80
driver118a
```

```
SOCKET_NETWORK socket3754 80
driver118b
SOCKET_NETWORK socket3755 80
driver118c
SOCKET_NETWORK socket3756 80
driver118d
SOCKET_NETWORK socket3757 80
driver118a
SOCKET_NETWORK socket3758 80
driver118b
SOCKET_NETWORK socket3759 80
driver118c
SOCKET_NETWORK socket3760 80
driver118d
SOCKET_NETWORK socket3761 80
driver118a
SOCKET_NETWORK socket3762 80
driver118b
SOCKET_NETWORK socket3763 80
driver118c
SOCKET_NETWORK socket3764 80
driver118d
SOCKET_NETWORK socket3765 80
driver118a
SOCKET_NETWORK socket3766 80
driver118b
SOCKET_NETWORK socket3767 80
driver118c
SOCKET_NETWORK socket3768 80
driver118d
SOCKET_NETWORK socket3769 80
driver118a
SOCKET_NETWORK socket3770 80
driver118b
SOCKET_NETWORK socket3771 80
driver118c
SOCKET_NETWORK socket3772 80
driver118d
SOCKET_NETWORK socket3773 80
driver118a
SOCKET_NETWORK socket3774 80
driver118b
SOCKET_NETWORK socket3775 80
driver118c
SOCKET_NETWORK socket3776 80
driver118d
SOCKET_NETWORK socket3777 80
driver119a
SOCKET_NETWORK socket3778 80
driver119b
SOCKET_NETWORK socket3779 80
driver119c
SOCKET_NETWORK socket3780 80
driver119d
SOCKET_NETWORK socket3781 80
driver119a
SOCKET_NETWORK socket3782 80
driver119b
SOCKET_NETWORK socket3783 80
driver119c
SOCKET_NETWORK socket3784 80
driver119d
SOCKET_NETWORK socket3785 80
driver119a
SOCKET_NETWORK socket3786 80
driver119b
SOCKET_NETWORK socket3787 80
driver119c
SOCKET_NETWORK socket3788 80
driver119d
SOCKET_NETWORK socket3789 80
driver119a
SOCKET_NETWORK socket3790 80
driver119b
SOCKET_NETWORK socket3791 80
driver119c
SOCKET_NETWORK socket3792 80
driver119d
SOCKET_NETWORK socket3793 80
driver119a
SOCKET_NETWORK socket3794 80
driver119b
SOCKET_NETWORK socket3795 80
driver119c
SOCKET_NETWORK socket3796 80
driver119d
SOCKET_NETWORK socket3797 80
driver119a
SOCKET_NETWORK socket3798 80
driver119b
SOCKET_NETWORK socket3799 80
driver119c
SOCKET_NETWORK socket3800 80
driver119d
SOCKET_NETWORK socket3801 80
driver119a
SOCKET_NETWORK socket3802 80
driver119b
SOCKET_NETWORK socket3803 80
driver119c
SOCKET_NETWORK socket3804 80
driver119d
SOCKET_NETWORK socket3805 80
driver119a
SOCKET_NETWORK socket3806 80
driver119b
SOCKET_NETWORK socket3807 80
driver119c
SOCKET_NETWORK socket3808 80
driver119d
SOCKET_NETWORK socket3809 80
driver120a
SOCKET_NETWORK socket3810 80
driver120b
SOCKET_NETWORK socket3811 80
driver120c
SOCKET_NETWORK socket3812 80
driver120d
SOCKET_NETWORK socket3813 80
driver120a
SOCKET_NETWORK socket3814 80
driver120b
SOCKET_NETWORK socket3815 80
driver120c
SOCKET_NETWORK socket3816 80
driver120d
SOCKET_NETWORK socket3817 80
driver120a
SOCKET_NETWORK socket3818 80
driver120b
SOCKET_NETWORK socket3819 80
driver120c
SOCKET_NETWORK socket3820 80
driver120d
SOCKET_NETWORK socket3821 80
driver120a
SOCKET_NETWORK socket3822 80
driver120b
SOCKET_NETWORK socket3823 80
driver120c
SOCKET_NETWORK socket3824 80
driver120d
SOCKET_NETWORK socket3825 80
driver120a
SOCKET_NETWORK socket3826 80
driver120b
SOCKET_NETWORK socket3827 80
driver120c
SOCKET_NETWORK socket3828 80
driver120d
SOCKET_NETWORK socket3829 80
driver120a
SOCKET_NETWORK socket3830 80
driver120b
SOCKET_NETWORK socket3831 80
driver120c
SOCKET_NETWORK socket3832 80
driver120d
SOCKET_NETWORK socket3833 80
driver120a
SOCKET_NETWORK socket3834 80
driver120b
SOCKET_NETWORK socket3835 80
driver120c
SOCKET_NETWORK socket3836 80
driver120d
SOCKET_NETWORK socket3837 80
driver120a
SOCKET_NETWORK socket3838 80
driver120b
SOCKET_NETWORK socket3839 80
driver120c
SOCKET_NETWORK socket3840 80
driver120d
#elif MASTER_NUM31
SOCKET_NETWORK socket3841 80
driver121a
SOCKET_NETWORK socket3842 80
driver121b
SOCKET_NETWORK socket3843 80
driver121c
SOCKET_NETWORK socket3844 80
driver121d
SOCKET_NETWORK socket3845 80
driver121a
SOCKET_NETWORK socket3846 80
driver121b
SOCKET_NETWORK socket3847 80
driver121c
SOCKET_NETWORK socket3848 80
driver121d
SOCKET_NETWORK socket3849 80
driver121a
SOCKET_NETWORK socket3850 80
driver121b
SOCKET_NETWORK socket3851 80
driver121c
SOCKET_NETWORK socket3852 80
driver121d
SOCKET_NETWORK socket3853 80
driver121a
SOCKET_NETWORK socket3854 80
driver121b
SOCKET_NETWORK socket3855 80
driver121c
SOCKET_NETWORK socket3856 80
driver121d
SOCKET_NETWORK socket3857 80
driver121a
SOCKET_NETWORK socket3858 80
driver121b
SOCKET_NETWORK socket3859 80
driver121c
SOCKET_NETWORK socket3860 80
driver121d
SOCKET_NETWORK socket3861 80
driver121a
SOCKET_NETWORK socket3862 80
driver121b
SOCKET_NETWORK socket3863 80
driver121c
SOCKET_NETWORK socket3864 80
driver121d
SOCKET_NETWORK socket3865 80
driver121a
SOCKET_NETWORK socket3866 80
driver121b
SOCKET_NETWORK socket3867 80
driver121c
SOCKET_NETWORK socket3868 80
driver121d
SOCKET_NETWORK socket3869 80
driver121a
SOCKET_NETWORK socket3870 80
driver121b
SOCKET_NETWORK socket3871 80
driver121c
SOCKET_NETWORK socket3872 80
driver121d
SOCKET_NETWORK socket3873 80
driver122a
SOCKET_NETWORK socket3874 80
driver122b
SOCKET_NETWORK socket3875 80
driver122c
SOCKET_NETWORK socket3876 80
driver122d
```

```
SOCKET_NETWORK socket3877 80
driver122a
SOCKET_NETWORK socket3878 80
driver122b
SOCKET_NETWORK socket3879 80
driver122c
SOCKET_NETWORK socket3880 80
driver122d
SOCKET_NETWORK socket3881 80
driver122a
SOCKET_NETWORK socket3882 80
driver122b
SOCKET_NETWORK socket3883 80
driver122c
SOCKET_NETWORK socket3884 80
driver122d
SOCKET_NETWORK socket3885 80
driver122a
SOCKET_NETWORK socket3886 80
driver122b
SOCKET_NETWORK socket3887 80
driver122c
SOCKET_NETWORK socket3888 80
driver122d
SOCKET_NETWORK socket3889 80
driver122a
SOCKET_NETWORK socket3890 80
driver122b
SOCKET_NETWORK socket3891 80
driver122c
SOCKET_NETWORK socket3892 80
driver122d
SOCKET_NETWORK socket3893 80
driver122a
SOCKET_NETWORK socket3894 80
driver122b
SOCKET_NETWORK socket3895 80
driver122c
SOCKET_NETWORK socket3896 80
driver122d
SOCKET_NETWORK socket3897 80
driver122a
SOCKET_NETWORK socket3898 80
driver122b
SOCKET_NETWORK socket3899 80
driver122c
SOCKET_NETWORK socket3900 80
driver122d
SOCKET_NETWORK socket3901 80
driver122a
SOCKET_NETWORK socket3902 80
driver122b
SOCKET_NETWORK socket3903 80
driver122c
SOCKET_NETWORK socket3904 80
driver122d
SOCKET_NETWORK socket3905 80
driver123a
SOCKET_NETWORK socket3906 80
driver123b
SOCKET_NETWORK socket3907 80
driver123c
SOCKET_NETWORK socket3908 80
driver123d
SOCKET_NETWORK socket3909 80
driver123a
SOCKET_NETWORK socket3910 80
driver123b
SOCKET_NETWORK socket3911 80
driver123c
SOCKET_NETWORK socket3912 80
driver123d
SOCKET_NETWORK socket3913 80
driver123a
SOCKET_NETWORK socket3914 80
driver123b
SOCKET_NETWORK socket3915 80
driver123c
SOCKET_NETWORK socket3916 80
driver123d
SOCKET_NETWORK socket3917 80
driver123a
SOCKET_NETWORK socket3918 80
driver123b
SOCKET_NETWORK socket3919 80
driver123c
SOCKET_NETWORK socket3920 80
driver123d
SOCKET_NETWORK socket3921 80
driver123a
SOCKET_NETWORK socket3922 80
driver123b
SOCKET_NETWORK socket3923 80
driver123c
SOCKET_NETWORK socket3924 80
driver123d
SOCKET_NETWORK socket3925 80
driver123a
SOCKET_NETWORK socket3926 80
driver123b
SOCKET_NETWORK socket3927 80
driver123c
SOCKET_NETWORK socket3928 80
driver123d
SOCKET_NETWORK socket3929 80
driver123a
SOCKET_NETWORK socket3930 80
driver123b
SOCKET_NETWORK socket3931 80
driver123c
SOCKET_NETWORK socket3932 80
driver123d
SOCKET_NETWORK socket3933 80
driver123a
SOCKET_NETWORK socket3934 80
driver123b
SOCKET_NETWORK socket3935 80
driver123c
SOCKET_NETWORK socket3936 80
driver123d
SOCKET_NETWORK socket3937 80
driver124a
SOCKET_NETWORK socket3938 80
driver124b
SOCKET_NETWORK socket3939 80
driver124c
SOCKET_NETWORK socket3940 80
driver124d
SOCKET_NETWORK socket3941 80
driver124a
SOCKET_NETWORK socket3942 80
driver124b
SOCKET_NETWORK socket3943 80
driver124c
SOCKET_NETWORK socket3944 80
driver124d
SOCKET_NETWORK socket3945 80
driver124a
SOCKET_NETWORK socket3946 80
driver124b
SOCKET_NETWORK socket3947 80
driver124c
SOCKET_NETWORK socket3948 80
driver124d
SOCKET_NETWORK socket3949 80
driver124a
SOCKET_NETWORK socket3950 80
driver124b
SOCKET_NETWORK socket3951 80
driver124c
SOCKET_NETWORK socket3952 80
driver124d
SOCKET_NETWORK socket3953 80
driver124a
SOCKET_NETWORK socket3954 80
driver124b
SOCKET_NETWORK socket3955 80
driver124c
SOCKET_NETWORK socket3956 80
driver124d
SOCKET_NETWORK socket3957 80
driver124a
SOCKET_NETWORK socket3958 80
driver124b
SOCKET_NETWORK socket3959 80
driver124c
SOCKET_NETWORK socket3960 80
driver124d
SOCKET_NETWORK socket3961 80
driver124a
SOCKET_NETWORK socket3962 80
driver124b
SOCKET_NETWORK socket3963 80
driver124c
SOCKET_NETWORK socket3964 80
driver124d
SOCKET_NETWORK socket3965 80
driver124a
SOCKET_NETWORK socket3966 80
driver124b
SOCKET_NETWORK socket3967 80
driver124c
SOCKET_NETWORK socket3968 80
driver124d
#elif MASTER_NUM32
SOCKET_NETWORK socket3969 80
driver125a
SOCKET_NETWORK socket3970 80
driver125b
SOCKET_NETWORK socket3971 80
driver125c
SOCKET_NETWORK socket3972 80
driver125d
SOCKET_NETWORK socket3973 80
driver125a
SOCKET_NETWORK socket3974 80
driver125b
SOCKET_NETWORK socket3975 80
driver125c
SOCKET_NETWORK socket3976 80
driver125d
SOCKET_NETWORK socket3977 80
driver125a
SOCKET_NETWORK socket3978 80
driver125b
SOCKET_NETWORK socket3979 80
driver125c
SOCKET_NETWORK socket3980 80
driver125d
SOCKET_NETWORK socket3981 80
driver125a
SOCKET_NETWORK socket3982 80
driver125b
SOCKET_NETWORK socket3983 80
driver125c
SOCKET_NETWORK socket3984 80
driver125d
SOCKET_NETWORK socket3985 80
driver125a
SOCKET_NETWORK socket3986 80
driver125b
SOCKET_NETWORK socket3987 80
driver125c
SOCKET_NETWORK socket3988 80
driver125d
SOCKET_NETWORK socket3989 80
driver125a
SOCKET_NETWORK socket3990 80
driver125b
SOCKET_NETWORK socket3991 80
driver125c
SOCKET_NETWORK socket3992 80
driver125d
SOCKET_NETWORK socket3993 80
driver125a
SOCKET_NETWORK socket3994 80
driver125b
SOCKET_NETWORK socket3995 80
driver125c
SOCKET_NETWORK socket3996 80
driver125d
SOCKET_NETWORK socket3997 80
driver125a
SOCKET_NETWORK socket3998 80
driver125b
SOCKET_NETWORK socket3999 80
driver125c
```

```
SOCKET_NETWORK socket4000   80
driver125d
SOCKET_NETWORK socket4001   80
driver126a
SOCKET_NETWORK socket4002   80
driver126b
SOCKET_NETWORK socket4003   80
driver126c
SOCKET_NETWORK socket4004   80
driver126d
SOCKET_NETWORK socket4005   80
driver126a
SOCKET_NETWORK socket4006   80
driver126b
SOCKET_NETWORK socket4007   80
driver126c
SOCKET_NETWORK socket4008   80
driver126d
SOCKET_NETWORK socket4009   80
driver126a
SOCKET_NETWORK socket4010   80
driver126b
SOCKET_NETWORK socket4011   80
driver126c
SOCKET_NETWORK socket4012   80
driver126d
SOCKET_NETWORK socket4013   80
driver126a
SOCKET_NETWORK socket4014   80
driver126b
SOCKET_NETWORK socket4015   80
driver126c
SOCKET_NETWORK socket4016   80
driver126d
SOCKET_NETWORK socket4017   80
driver126a
SOCKET_NETWORK socket4018   80
driver126b
SOCKET_NETWORK socket4019   80
driver126c
SOCKET_NETWORK socket4020   80
driver126d
SOCKET_NETWORK socket4021   80
driver126a
SOCKET_NETWORK socket4022   80
driver126b
SOCKET_NETWORK socket4023   80
driver126c
SOCKET_NETWORK socket4024   80
driver126d
SOCKET_NETWORK socket4025   80
driver126a
SOCKET_NETWORK socket4026   80
driver126b
SOCKET_NETWORK socket4027   80
driver126c
SOCKET_NETWORK socket4028   80
driver126d
SOCKET_NETWORK socket4029   80
driver126a
SOCKET_NETWORK socket4030   80
driver126b
SOCKET_NETWORK socket4031   80
driver126c
SOCKET_NETWORK socket4032   80
driver126d
SOCKET_NETWORK socket4033   80
driver127a
SOCKET_NETWORK socket4034   80
driver127b
SOCKET_NETWORK socket4035   80
driver127c
SOCKET_NETWORK socket4036   80
driver127d
SOCKET_NETWORK socket4037   80
driver127a
SOCKET_NETWORK socket4038   80
driver127b
SOCKET_NETWORK socket4039   80
driver127c
SOCKET_NETWORK socket4040   80
driver127d
SOCKET_NETWORK socket4041   80
driver127a
SOCKET_NETWORK socket4042   80
driver127b
SOCKET_NETWORK socket4043   80
driver127c
SOCKET_NETWORK socket4044   80
driver127d
SOCKET_NETWORK socket4045   80
driver127a
SOCKET_NETWORK socket4046   80
driver127b
SOCKET_NETWORK socket4047   80
driver127c
SOCKET_NETWORK socket4048   80
driver127d
SOCKET_NETWORK socket4049   80
driver127a
SOCKET_NETWORK socket4050   80
driver127b
SOCKET_NETWORK socket4051   80
driver127c
SOCKET_NETWORK socket4052   80
driver127d
SOCKET_NETWORK socket4053   80
driver127a
SOCKET_NETWORK socket4054   80
driver127b
SOCKET_NETWORK socket4055   80
driver127c
SOCKET_NETWORK socket4056   80
driver127d
SOCKET_NETWORK socket4057   80
driver127a
SOCKET_NETWORK socket4058   80
driver127b
SOCKET_NETWORK socket4059   80
driver127c
SOCKET_NETWORK socket4060   80
driver127d
SOCKET_NETWORK socket4061   80
driver127a
SOCKET_NETWORK socket4062   80
driver127b
SOCKET_NETWORK socket4063   80
driver127c
SOCKET_NETWORK socket4064   80
driver127d
SOCKET_NETWORK socket4065   80
driver128a
SOCKET_NETWORK socket4066   80
driver128b
SOCKET_NETWORK socket4067   80
driver128c
SOCKET_NETWORK socket4068   80
driver128d
SOCKET_NETWORK socket4069   80
driver128a
SOCKET_NETWORK socket4070   80
driver128b
SOCKET_NETWORK socket4071   80
driver128c
SOCKET_NETWORK socket4072   80
driver128d
SOCKET_NETWORK socket4073   80
driver128a
SOCKET_NETWORK socket4074   80
driver128b
SOCKET_NETWORK socket4075   80
driver128c
SOCKET_NETWORK socket4076   80
driver128d
SOCKET_NETWORK socket4077   80
driver128a
SOCKET_NETWORK socket4078   80
driver128b
SOCKET_NETWORK socket4079   80
driver128c
SOCKET_NETWORK socket4080   80
driver128d
SOCKET_NETWORK socket4081   80
driver128a
SOCKET_NETWORK socket4082   80
driver128b
SOCKET_NETWORK socket4083   80
driver128c
SOCKET_NETWORK socket4084   80
driver128d
SOCKET_NETWORK socket4085   80
driver128a
SOCKET_NETWORK socket4086   80
driver128b
SOCKET_NETWORK socket4087   80
driver128c
SOCKET_NETWORK socket4088   80
driver128d
SOCKET_NETWORK socket4089   80
driver128a
SOCKET_NETWORK socket4090   80
driver128b
SOCKET_NETWORK socket4091   80
driver128c
SOCKET_NETWORK socket4092   80
driver128d
SOCKET_NETWORK socket4093   80
driver128a
SOCKET_NETWORK socket4094   80
driver128b
SOCKET_NETWORK socket4095   80
driver128c
SOCKET_NETWORK socket4096   80
driver128d
#endif
/*-----------------------------------------------------------
-------*/
OUTPUTNAME="regattaH"
CPU=48
#if 0
BEGIN_WAIT=5:00
RAMPUP=20:30
RUNTIME=30:00
RAMPDOWN_WAIT=5:00
RAMPDOWN=20:00
#else
BEGIN_WAIT=25:00
RAMPUP=45:00
RUNTIME=150:00
RAMPDOWN_WAIT=2:00
RAMPDOWN=20:00
#endif
/*RAMPUP_SEC 2700 */
/*RUNTIME_SEC 9000  */
/*WAREHOUSES 256000 */
/*CLIENTS 3712 */
INTERVAL=1:00 /* Interval to calculate mix
from */
LOGIN_MAX_LOAD = 3
LOGIN_BEGIN = 0 /* skip login state if set
to 1 */
NOBEGIN = 1
KEYSTROKE_PACKET_SIZE = 0
MAX_CONCURRENT_SPAWN = 3
SPAWN_COUNT = 3
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 620 0-62
START_RANGE client1a socket2 630 62-125
START_RANGE client1a socket3 620 125-187
START_RANGE client1a socket4 630 187-250
START_RANGE client1a socket5 620 250-312
START_RANGE client1a socket6 630 312-375
START_RANGE client1a socket7 620 375-437
START_RANGE client1a socket8 630 437-500
START_RANGE client1a socket9 620 500-562
START_RANGE client1a socket10 630 562-625
START_RANGE client1a socket11 620 625-687
START_RANGE client1a socket12 630 687-750
START_RANGE client1a socket13 620 750-812
START_RANGE client1a socket14 630 812-875
START_RANGE client1a socket15 620 875-937
START_RANGE client1a socket16 630 937-1000
START_RANGE client1a socket17 620 1000-1062
START_RANGE client1a socket18 630 1062-1125
START_RANGE client1a socket19 620 1125-1187
START_RANGE client1a socket20 630 1187-1250
START_RANGE client1a socket21 620 1250-1312
START_RANGE client1a socket22 630 1312-1375
START_RANGE client1a socket23 620 1375-1437
START_RANGE client1a socket24 630 1437-1500
START_RANGE client1a socket25 620 1500-1562
START_RANGE client1a socket26 630 1562-1625
START_RANGE client1a socket27 620 1625-1687
START_RANGE client1a socket28 630 1687-1750
START_RANGE client1a socket29 620 1750-1812
START_RANGE client1a socket30 630 1812-1875
START_RANGE client1a socket31 620 1875-1937
START_RANGE client1a socket32 630 1937-2000
START_RANGE client2a socket33 620 2000-2062
START_RANGE client2a socket34 630 2062-2125
START_RANGE client2a socket35 620 2125-2187
START_RANGE client2a socket36 630 2187-2250
START_RANGE client2a socket37 620 2250-2312
START_RANGE client2a socket38 630 2312-2375
START_RANGE client2a socket39 620 2375-2437
START_RANGE client2a socket40 630 2437-2500
START_RANGE client2a socket41 620 2500-2562
START_RANGE client2a socket42 630 2562-2625
START_RANGE client2a socket43 620 2625-2687
START_RANGE client2a socket44 630 2687-2750
START_RANGE client2a socket45 620 2750-2812
START_RANGE client2a socket46 630 2812-2875
START_RANGE client2a socket47 620 2875-2937
START_RANGE client2a socket48 630 2937-3000
START_RANGE client2a socket49 620 3000-3062
START_RANGE client2a socket50 630 3062-3125
START_RANGE client2a socket51 620 3125-3187
START_RANGE client2a socket52 630 3187-3250
START_RANGE client2a socket53 620 3250-3312
START_RANGE client2a socket54 630 3312-3375
START_RANGE client2a socket55 620 3375-3437
START_RANGE client2a socket56 630 3437-3500
START_RANGE client2a socket57 620 3500-3562
START_RANGE client2a socket58 630 3562-3625
START_RANGE client2a socket59 620 3625-3687
START_RANGE client2a socket60 630 3687-3750
START_RANGE client2a socket61 620 3750-3812
START_RANGE client2a socket62 630 3812-3875
START_RANGE client2a socket63 620 3875-3937
START_RANGE client2a socket64 630 3937-4000
START_RANGE client3a socket65 620 4000-4062
START_RANGE client3a socket66 630 4062-4125
START_RANGE client3a socket67 620 4125-4187
START_RANGE client3a socket68 630 4187-4250
START_RANGE client3a socket69 620 4250-4312
START_RANGE client3a socket70 630 4312-4375
START_RANGE client3a socket71 620 4375-4437
START_RANGE client3a socket72 630 4437-4500
START_RANGE client3a socket73 620 4500-4562
START_RANGE client3a socket74 630 4562-4625
START_RANGE client3a socket75 620 4625-4687
START_RANGE client3a socket76 630 4687-4750
START_RANGE client3a socket77 620 4750-4812
START_RANGE client3a socket78 630 4812-4875
START_RANGE client3a socket79 620 4875-4937
START_RANGE client3a socket80 630 4937-5000
START_RANGE client3a socket81 620 5000-5062
START_RANGE client3a socket82 630 5062-5125
START_RANGE client3a socket83 620 5125-5187
START_RANGE client3a socket84 630 5187-5250
START_RANGE client3a socket85 620 5250-5312
START_RANGE client3a socket86 630 5312-5375
START_RANGE client3a socket87 620 5375-5437
START_RANGE client3a socket88 630 5437-5500
START_RANGE client3a socket89 620 5500-5562
START_RANGE client3a socket90 630 5562-5625
START_RANGE client3a socket91 620 5625-5687
START_RANGE client3a socket92 630 5687-5750
START_RANGE client3a socket93 620 5750-5812
START_RANGE client3a socket94 630 5812-5875
START_RANGE client3a socket95 620 5875-5937
START_RANGE client3a socket96 630 5937-6000
START_RANGE client4a socket97 620 6000-6062
START_RANGE client4a socket98 630 6062-6125
START_RANGE client4a socket99 620 6125-6187
START_RANGE client4a socket100 630 6187-6250
START_RANGE client4a socket101 620 6250-6312
START_RANGE client4a socket102 630 6312-6375
START_RANGE client4a socket103 620 6375-6437
START_RANGE client4a socket104 630 6437-6500
START_RANGE client4a socket105 620 6500-6562
START_RANGE client4a socket106 630 6562-6625
START_RANGE client4a socket107 620 6625-6687
START_RANGE client4a socket108 630 6687-6750
START_RANGE client4a socket109 620 6750-6812
START_RANGE client4a socket110 630 6812-6875
START_RANGE client4a socket111 620 6875-6937
START_RANGE client4a socket112 630 6937-7000
START_RANGE client4a socket113 620 7000-7062
START_RANGE client4a socket114 630 7062-7125
START_RANGE client4a socket115 620 7125-7187
START_RANGE client4a socket116 630 7187-7250
START_RANGE client4a socket117 620 7250-7312
```

```
START_RANGE client4a socket118 630
7312-7375
START_RANGE client4a socket119 620
7375-7437
START_RANGE client4a socket120 630
7437-7500
START_RANGE client4a socket121 620
7500-7562
START_RANGE client4a socket122 630
7562-7625
START_RANGE client4a socket123 620
7625-7687
START_RANGE client4a socket124 630
7687-7750
START_RANGE client4a socket125 620
7750-7812
START_RANGE client4a socket126 630
7812-7875
START_RANGE client4a socket127 620
7875-7937
START_RANGE client4a socket128 630
7937-8000
#elif MASTER_NUM2
START_RANGE client5a socket129 620
8000-8062
START_RANGE client5a socket130 630
8062-8125
START_RANGE client5a socket131 620
8125-8187
START_RANGE client5a socket132 630
8187-8250
START_RANGE client5a socket133 620
8250-8312
START_RANGE client5a socket134 630
8312-8375
START_RANGE client5a socket135 620
8375-8437
START_RANGE client5a socket136 630
8437-8500
START_RANGE client5a socket137 620
8500-8562
START_RANGE client5a socket138 630
8562-8625
START_RANGE client5a socket139 620
8625-8687
START_RANGE client5a socket140 630
8687-8750
START_RANGE client5a socket141 620
8750-8812
START_RANGE client5a socket142 630
8812-8875
START_RANGE client5a socket143 620
8875-8937
START_RANGE client5a socket144 630
8937-9000
START_RANGE client5a socket145 620
9000-9062
START_RANGE client5a socket146 630
9062-9125
START_RANGE client5a socket147 620
9125-9187
START_RANGE client5a socket148 630
9187-9250
START_RANGE client5a socket149 620
9250-9312
START_RANGE client5a socket150 630
9312-9375
START_RANGE client5a socket151 620
9375-9437
START_RANGE client5a socket152 630
9437-9500
START_RANGE client5a socket153 620
9500-9562
START_RANGE client5a socket154 630
9562-9625
START_RANGE client5a socket155 620
9625-9687
START_RANGE client5a socket156 630
9687-9750
START_RANGE client5a socket157 620
9750-9812
START_RANGE client5a socket158 630
9812-9875
START_RANGE client5a socket159 620
9875-9937
START_RANGE client5a socket160 630
9937-10000
START_RANGE client6a socket161 620
10000-10062
START_RANGE client6a socket162 630
10062-10125
START_RANGE client6a socket163 620
10125-10187
START_RANGE client6a socket164 630
10187-10250
START_RANGE client6a socket165 620
10250-10312
START_RANGE client6a socket166 630
10312-10375
START_RANGE client6a socket167 620
10375-10437
START_RANGE client6a socket168 630
10437-10500
START_RANGE client6a socket169 620
10500-10562
START_RANGE client6a socket170 630
10562-10625
START_RANGE client6a socket171 620
10625-10687
START_RANGE client6a socket172 630
10687-10750
START_RANGE client6a socket173 620
10750-10812
START_RANGE client6a socket174 630
10812-10875
START_RANGE client6a socket175 620
10875-10937
START_RANGE client6a socket176 630
10937-11000
START_RANGE client6a socket177 620
11000-11062
START_RANGE client6a socket178 630
11062-11125
START_RANGE client6a socket179 620
11125-11187
START_RANGE client6a socket180 630
11187-11250
START_RANGE client6a socket181 620
11250-11312
START_RANGE client6a socket182 630
11312-11375
START_RANGE client6a socket183 620
11375-11437
START_RANGE client6a socket184 630
11437-11500
START_RANGE client6a socket185 620
11500-11562
START_RANGE client6a socket186 630
11562-11625
START_RANGE client6a socket187 620
11625-11687
START_RANGE client6a socket188 630
11687-11750
START_RANGE client6a socket189 620
11750-11812
START_RANGE client6a socket190 630
11812-11875
START_RANGE client6a socket191 620
11875-11937
START_RANGE client6a socket192 630
11937-12000
START_RANGE client7a socket193 620
12000-12062
START_RANGE client7a socket194 630
12062-12125
START_RANGE client7a socket195 620
12125-12187
START_RANGE client7a socket196 630
12187-12250
START_RANGE client7a socket197 620
12250-12312
START_RANGE client7a socket198 630
12312-12375
START_RANGE client7a socket199 620
12375-12437
START_RANGE client7a socket200 630
12437-12500
START_RANGE client7a socket201 620
12500-12562
START_RANGE client7a socket202 630
12562-12625
START_RANGE client7a socket203 620
12625-12687
START_RANGE client7a socket204 630
12687-12750
START_RANGE client7a socket205 620
12750-12812
START_RANGE client7a socket206 630
12812-12875
START_RANGE client7a socket207 620
12875-12937
START_RANGE client7a socket208 630
12937-13000
START_RANGE client7a socket209 620
13000-13062
START_RANGE client7a socket210 630
13062-13125
START_RANGE client7a socket211 620
13125-13187
START_RANGE client7a socket212 630
13187-13250
START_RANGE client7a socket213 620
13250-13312
START_RANGE client7a socket214 630
13312-13375
START_RANGE client7a socket215 620
13375-13437
START_RANGE client7a socket216 630
13437-13500
START_RANGE client7a socket217 620
13500-13562
START_RANGE client7a socket218 630
13562-13625
START_RANGE client7a socket219 620
13625-13687
START_RANGE client7a socket220 630
13687-13750
START_RANGE client7a socket221 620
13750-13812
START_RANGE client7a socket222 630
13812-13875
START_RANGE client7a socket223 620
13875-13937
START_RANGE client7a socket224 630
13937-14000
START_RANGE client8a socket225 620
14000-14062
START_RANGE client8a socket226 630
14062-14125
START_RANGE client8a socket227 620
14125-14187
START_RANGE client8a socket228 630
14187-14250
START_RANGE client8a socket229 620
14250-14312
START_RANGE client8a socket230 630
14312-14375
START_RANGE client8a socket231 620
14375-14437
START_RANGE client8a socket232 630
14437-14500
START_RANGE client8a socket233 620
14500-14562
START_RANGE client8a socket234 630
14562-14625
START_RANGE client8a socket235 620
14625-14687
START_RANGE client8a socket236 630
14687-14750
START_RANGE client8a socket237 620
14750-14812
START_RANGE client8a socket238 630
14812-14875
START_RANGE client8a socket239 620
14875-14937
START_RANGE client8a socket240 630
14937-15000
```

```
START_RANGE client8a socket241 620
15000-15062
START_RANGE client8a socket242 630
15062-15125
START_RANGE client8a socket243 620
15125-15187
START_RANGE client8a socket244 630
15187-15250
START_RANGE client8a socket245 620
15250-15312
START_RANGE client8a socket246 630
15312-15375
START_RANGE client8a socket247 620
15375-15437
START_RANGE client8a socket248 630
15437-15500
START_RANGE client8a socket249 620
15500-15562
START_RANGE client8a socket250 630
15562-15625
START_RANGE client8a socket251 620
15625-15687
START_RANGE client8a socket252 630
15687-15750
START_RANGE client8a socket253 620
15750-15812
START_RANGE client8a socket254 630
15812-15875
START_RANGE client8a socket255 620
15875-15937
START_RANGE client8a socket256 630
15937-16000
#elif MASTER_NUM3
START_RANGE client9a socket257 620
16000-16062
START_RANGE client9a socket258 630
16062-16125
START_RANGE client9a socket259 620
16125-16187
START_RANGE client9a socket260 630
16187-16250
START_RANGE client9a socket261 620
16250-16312
START_RANGE client9a socket262 630
16312-16375
START_RANGE client9a socket263 620
16375-16437
START_RANGE client9a socket264 630
16437-16500
START_RANGE client9a socket265 620
16500-16562
START_RANGE client9a socket266 630
16562-16625
START_RANGE client9a socket267 620
16625-16687
START_RANGE client9a socket268 630
16687-16750
START_RANGE client9a socket269 620
16750-16812
START_RANGE client9a socket270 630
16812-16875
START_RANGE client9a socket271 620
16875-16937
START_RANGE client9a socket272 630
16937-17000
START_RANGE client9a socket273 620
17000-17062
START_RANGE client9a socket274 630
17062-17125
START_RANGE client9a socket275 620
17125-17187
START_RANGE client9a socket276 630
17187-17250
START_RANGE client9a socket277 620
17250-17312
START_RANGE client9a socket278 630
17312-17375
START_RANGE client9a socket279 620
17375-17437
START_RANGE client9a socket280 630
17437-17500
START_RANGE client9a socket281 620
17500-17562
START_RANGE client9a socket282 630
17562-17625
START_RANGE client9a socket283 620
17625-17687
START_RANGE client9a socket284 630
17687-17750
START_RANGE client9a socket285 620
17750-17812
START_RANGE client9a socket286 630
17812-17875
START_RANGE client9a socket287 620
17875-17937
START_RANGE client9a socket288 630
17937-18000
START_RANGE client10a socket289 620
18000-18062
START_RANGE client10a socket290 630
18062-18125
START_RANGE client10a socket291 620
18125-18187
START_RANGE client10a socket292 630
18187-18250
START_RANGE client10a socket293 620
18250-18312
START_RANGE client10a socket294 630
18312-18375
START_RANGE client10a socket295 620
18375-18437
START_RANGE client10a socket296 630
18437-18500
START_RANGE client10a socket297 620
18500-18562
START_RANGE client10a socket298 630
18562-18625
START_RANGE client10a socket299 620
18625-18687
START_RANGE client10a socket300 630
18687-18750
START_RANGE client10a socket301 620
18750-18812
START_RANGE client10a socket302 630
18812-18875
START_RANGE client10a socket303 620
18875-18937
START_RANGE client10a socket304 630
18937-19000
START_RANGE client10a socket305 620
19000-19062
START_RANGE client10a socket306 630
19062-19125
START_RANGE client10a socket307 620
19125-19187
START_RANGE client10a socket308 630
19187-19250
START_RANGE client10a socket309 620
19250-19312
START_RANGE client10a socket310 630
19312-19375
START_RANGE client10a socket311 620
19375-19437
START_RANGE client10a socket312 630
19437-19500
START_RANGE client10a socket313 620
19500-19562
START_RANGE client10a socket314 630
19562-19625
START_RANGE client10a socket315 620
19625-19687
START_RANGE client10a socket316 630
19687-19750
START_RANGE client10a socket317 620
19750-19812
START_RANGE client10a socket318 630
19812-19875
START_RANGE client10a socket319 620
19875-19937
START_RANGE client10a socket320 630
19937-20000
START_RANGE client11a socket321 620
20000-20062
START_RANGE client11a socket322 630
20062-20125
START_RANGE client11a socket323 620
20125-20187
START_RANGE client11a socket324 630
20187-20250
START_RANGE client11a socket325 620
20250-20312
START_RANGE client11a socket326 630
20312-20375
START_RANGE client11a socket327 620
20375-20437
START_RANGE client11a socket328 630
20437-20500
START_RANGE client11a socket329 620
20500-20562
START_RANGE client11a socket330 630
20562-20625
START_RANGE client11a socket331 620
20625-20687
START_RANGE client11a socket332 630
20687-20750
START_RANGE client11a socket333 620
20750-20812
START_RANGE client11a socket334 630
20812-20875
START_RANGE client11a socket335 620
20875-20937
START_RANGE client11a socket336 630
20937-21000
START_RANGE client11a socket337 620
21000-21062
START_RANGE client11a socket338 630
21062-21125
START_RANGE client11a socket339 620
21125-21187
START_RANGE client11a socket340 630
21187-21250
START_RANGE client11a socket341 620
21250-21312
START_RANGE client11a socket342 630
21312-21375
START_RANGE client11a socket343 620
21375-21437
START_RANGE client11a socket344 630
21437-21500
START_RANGE client11a socket345 620
21500-21562
START_RANGE client11a socket346 630
21562-21625
START_RANGE client11a socket347 620
21625-21687
START_RANGE client11a socket348 630
21687-21750
START_RANGE client11a socket349 620
21750-21812
START_RANGE client11a socket350 630
21812-21875
START_RANGE client11a socket351 620
21875-21937
START_RANGE client11a socket352 630
21937-22000
START_RANGE client12a socket353 620
22000-22062
START_RANGE client12a socket354 630
22062-22125
START_RANGE client12a socket355 620
22125-22187
START_RANGE client12a socket356 630
22187-22250
START_RANGE client12a socket357 620
22250-22312
START_RANGE client12a socket358 630
22312-22375
START_RANGE client12a socket359 620
22375-22437
START_RANGE client12a socket360 630
22437-22500
START_RANGE client12a socket361 620
22500-22562
START_RANGE client12a socket362 630
22562-22625
START_RANGE client12a socket363 620
22625-22687
```

```
START_RANGE client12a socket364 630
22687-22750
START_RANGE client12a socket365 620
22750-22812
START_RANGE client12a socket366 630
22812-22875
START_RANGE client12a socket367 620
22875-22937
START_RANGE client12a socket368 630
22937-23000
START_RANGE client12a socket369 620
23000-23062
START_RANGE client12a socket370 630
23062-23125
START_RANGE client12a socket371 620
23125-23187
START_RANGE client12a socket372 630
23187-23250
START_RANGE client12a socket373 620
23250-23312
START_RANGE client12a socket374 630
23312-23375
START_RANGE client12a socket375 620
23375-23437
START_RANGE client12a socket376 630
23437-23500
START_RANGE client12a socket377 620
23500-23562
START_RANGE client12a socket378 630
23562-23625
START_RANGE client12a socket379 620
23625-23687
START_RANGE client12a socket380 630
23687-23750
START_RANGE client12a socket381 620
23750-23812
START_RANGE client12a socket382 630
23812-23875
START_RANGE client12a socket383 620
23875-23937
START_RANGE client12a socket384 630
23937-24000
#elif MASTER_NUM4
START_RANGE client13a socket385 620
24000-24062
START_RANGE client13a socket386 630
24062-24125
START_RANGE client13a socket387 620
24125-24187
START_RANGE client13a socket388 630
24187-24250
START_RANGE client13a socket389 620
24250-24312
START_RANGE client13a socket390 630
24312-24375
START_RANGE client13a socket391 620
24375-24437
START_RANGE client13a socket392 630
24437-24500
START_RANGE client13a socket393 620
24500-24562
START_RANGE client13a socket394 630
24562-24625
START_RANGE client13a socket395 620
24625-24687
START_RANGE client13a socket396 630
24687-24750
START_RANGE client13a socket397 620
24750-24812
START_RANGE client13a socket398 630
24812-24875
START_RANGE client13a socket399 620
24875-24937
START_RANGE client13a socket400 630
24937-25000
START_RANGE client13a socket401 620
25000-25062
START_RANGE client13a socket402 630
25062-25125
START_RANGE client13a socket403 620
25125-25187
START_RANGE client13a socket404 630
25187-25250
START_RANGE client13a socket405 620
25250-25312
START_RANGE client13a socket406 630
25312-25375
START_RANGE client13a socket407 620
25375-25437
START_RANGE client13a socket408 630
25437-25500
START_RANGE client13a socket409 620
25500-25562
START_RANGE client13a socket410 630
25562-25625
START_RANGE client13a socket411 620
25625-25687
START_RANGE client13a socket412 630
25687-25750
START_RANGE client13a socket413 620
25750-25812
START_RANGE client13a socket414 630
25812-25875
START_RANGE client13a socket415 620
25875-25937
START_RANGE client13a socket416 630
25937-26000
START_RANGE client14a socket417 620
26000-26062
START_RANGE client14a socket418 630
26062-26125
START_RANGE client14a socket419 620
26125-26187
START_RANGE client14a socket420 630
26187-26250
START_RANGE client14a socket421 620
26250-26312
START_RANGE client14a socket422 630
26312-26375
START_RANGE client14a socket423 620
26375-26437
START_RANGE client14a socket424 630
26437-26500
START_RANGE client14a socket425 620
26500-26562
START_RANGE client14a socket426 630
26562-26625
START_RANGE client14a socket427 620
26625-26687
START_RANGE client14a socket428 630
26687-26750
START_RANGE client14a socket429 620
26750-26812
START_RANGE client14a socket430 630
26812-26875
START_RANGE client14a socket431 620
26875-26937
START_RANGE client14a socket432 630
26937-27000
START_RANGE client14a socket433 620
27000-27062
START_RANGE client14a socket434 630
27062-27125
START_RANGE client14a socket435 620
27125-27187
START_RANGE client14a socket436 630
27187-27250
START_RANGE client14a socket437 620
27250-27312
START_RANGE client14a socket438 630
27312-27375
START_RANGE client14a socket439 620
27375-27437
START_RANGE client14a socket440 630
27437-27500
START_RANGE client14a socket441 620
27500-27562
START_RANGE client14a socket442 630
27562-27625
START_RANGE client14a socket443 620
27625-27687
START_RANGE client14a socket444 630
27687-27750
START_RANGE client14a socket445 620
27750-27812
START_RANGE client14a socket446 630
27812-27875
START_RANGE client14a socket447 620
27875-27937
START_RANGE client14a socket448 630
27937-28000
START_RANGE client15a socket449 620
28000-28062
START_RANGE client15a socket450 630
28062-28125
START_RANGE client15a socket451 620
28125-28187
START_RANGE client15a socket452 630
28187-28250
START_RANGE client15a socket453 620
28250-28312
START_RANGE client15a socket454 630
28312-28375
START_RANGE client15a socket455 620
28375-28437
START_RANGE client15a socket456 630
28437-28500
START_RANGE client15a socket457 620
28500-28562
START_RANGE client15a socket458 630
28562-28625
START_RANGE client15a socket459 620
28625-28687
START_RANGE client15a socket460 630
28687-28750
START_RANGE client15a socket461 620
28750-28812
START_RANGE client15a socket462 630
28812-28875
START_RANGE client15a socket463 620
28875-28937
START_RANGE client15a socket464 630
28937-29000
START_RANGE client15a socket465 620
29000-29062
START_RANGE client15a socket466 630
29062-29125
START_RANGE client15a socket467 620
29125-29187
START_RANGE client15a socket468 630
29187-29250
START_RANGE client15a socket469 620
29250-29312
START_RANGE client15a socket470 630
29312-29375
START_RANGE client15a socket471 620
29375-29437
START_RANGE client15a socket472 630
29437-29500
START_RANGE client15a socket473 620
29500-29562
START_RANGE client15a socket474 630
29562-29625
START_RANGE client15a socket475 620
29625-29687
START_RANGE client15a socket476 630
29687-29750
START_RANGE client15a socket477 620
29750-29812
START_RANGE client15a socket478 630
29812-29875
START_RANGE client15a socket479 620
29875-29937
START_RANGE client15a socket480 630
29937-30000
START_RANGE client16a socket481 620
30000-30062
START_RANGE client16a socket482 630
30062-30125
START_RANGE client16a socket483 620
30125-30187
START_RANGE client16a socket484 630
30187-30250
START_RANGE client16a socket485 620
30250-30312
START_RANGE client16a socket486 630
30312-30375
```

```
START_RANGE client16a socket487 620
30375-30437
START_RANGE client16a socket488 630
30437-30500
START_RANGE client16a socket489 620
30500-30562
START_RANGE client16a socket490 630
30562-30625
START_RANGE client16a socket491 620
30625-30687
START_RANGE client16a socket492 630
30687-30750
START_RANGE client16a socket493 620
30750-30812
START_RANGE client16a socket494 630
30812-30875
START_RANGE client16a socket495 620
30875-30937
START_RANGE client16a socket496 630
30937-31000
START_RANGE client16a socket497 620
31000-31062
START_RANGE client16a socket498 630
31062-31125
START_RANGE client16a socket499 620
31125-31187
START_RANGE client16a socket500 630
31187-31250
START_RANGE client16a socket501 620
31250-31312
START_RANGE client16a socket502 630
31312-31375
START_RANGE client16a socket503 620
31375-31437
START_RANGE client16a socket504 630
31437-31500
START_RANGE client16a socket505 620
31500-31562
START_RANGE client16a socket506 630
31562-31625
START_RANGE client16a socket507 620
31625-31687
START_RANGE client16a socket508 630
31687-31750
START_RANGE client16a socket509 620
31750-31812
START_RANGE client16a socket510 630
31812-31875
START_RANGE client16a socket511 620
31875-31937
START_RANGE client16a socket512 630
31937-32000
#elif MASTER_NUM5
START_RANGE client17a socket513 620
32000-32062
START_RANGE client17a socket514 630
32062-32125
START_RANGE client17a socket515 620
32125-32187
START_RANGE client17a socket516 630
32187-32250
START_RANGE client17a socket517 620
32250-32312
START_RANGE client17a socket518 630
32312-32375
START_RANGE client17a socket519 620
32375-32437
START_RANGE client17a socket520 630
32437-32500
START_RANGE client17a socket521 620
32500-32562
START_RANGE client17a socket522 630
32562-32625
START_RANGE client17a socket523 620
32625-32687
START_RANGE client17a socket524 630
32687-32750
START_RANGE client17a socket525 620
32750-32812
START_RANGE client17a socket526 630
32812-32875
START_RANGE client17a socket527 620
32875-32937
START_RANGE client17a socket528 630
32937-33000
START_RANGE client17a socket529 620
33000-33062
START_RANGE client17a socket530 630
33062-33125
START_RANGE client17a socket531 620
33125-33187
START_RANGE client17a socket532 630
33187-33250
START_RANGE client17a socket533 620
33250-33312
START_RANGE client17a socket534 630
33312-33375
START_RANGE client17a socket535 620
33375-33437
START_RANGE client17a socket536 630
33437-33500
START_RANGE client17a socket537 620
33500-33562
START_RANGE client17a socket538 630
33562-33625
START_RANGE client17a socket539 620
33625-33687
START_RANGE client17a socket540 630
33687-33750
START_RANGE client17a socket541 620
33750-33812
START_RANGE client17a socket542 630
33812-33875
START_RANGE client17a socket543 620
33875-33937
START_RANGE client17a socket544 630
33937-34000
START_RANGE client18a socket545 620
34000-34062
START_RANGE client18a socket546 630
34062-34125
START_RANGE client18a socket547 620
34125-34187
START_RANGE client18a socket548 630
34187-34250
START_RANGE client18a socket549 620
34250-34312
START_RANGE client18a socket550 630
34312-34375
START_RANGE client18a socket551 620
34375-34437
START_RANGE client18a socket552 630
34437-34500
START_RANGE client18a socket553 620
34500-34562
START_RANGE client18a socket554 630
34562-34625
START_RANGE client18a socket555 620
34625-34687
START_RANGE client18a socket556 630
34687-34750
START_RANGE client18a socket557 620
34750-34812
START_RANGE client18a socket558 630
34812-34875
START_RANGE client18a socket559 620
34875-34937
START_RANGE client18a socket560 630
34937-35000
START_RANGE client18a socket561 620
35000-35062
START_RANGE client18a socket562 630
35062-35125
START_RANGE client18a socket563 620
35125-35187
START_RANGE client18a socket564 630
35187-35250
START_RANGE client18a socket565 620
35250-35312
START_RANGE client18a socket566 630
35312-35375
START_RANGE client18a socket567 620
35375-35437
START_RANGE client18a socket568 630
35437-35500
START_RANGE client18a socket569 620
35500-35562
START_RANGE client18a socket570 630
35562-35625
START_RANGE client18a socket571 620
35625-35687
START_RANGE client18a socket572 630
35687-35750
START_RANGE client18a socket573 620
35750-35812
START_RANGE client18a socket574 630
35812-35875
START_RANGE client18a socket575 620
35875-35937
START_RANGE client18a socket576 630
35937-36000
START_RANGE client19a socket577 620
36000-36062
START_RANGE client19a socket578 630
36062-36125
START_RANGE client19a socket579 620
36125-36187
START_RANGE client19a socket580 630
36187-36250
START_RANGE client19a socket581 620
36250-36312
START_RANGE client19a socket582 630
36312-36375
START_RANGE client19a socket583 620
36375-36437
START_RANGE client19a socket584 630
36437-36500
START_RANGE client19a socket585 620
36500-36562
START_RANGE client19a socket586 630
36562-36625
START_RANGE client19a socket587 620
36625-36687
START_RANGE client19a socket588 630
36687-36750
START_RANGE client19a socket589 620
36750-36812
START_RANGE client19a socket590 630
36812-36875
START_RANGE client19a socket591 620
36875-36937
START_RANGE client19a socket592 630
36937-37000
START_RANGE client19a socket593 620
37000-37062
START_RANGE client19a socket594 630
37062-37125
START_RANGE client19a socket595 620
37125-37187
START_RANGE client19a socket596 630
37187-37250
START_RANGE client19a socket597 620
37250-37312
START_RANGE client19a socket598 630
37312-37375
START_RANGE client19a socket599 620
37375-37437
START_RANGE client19a socket600 630
37437-37500
START_RANGE client19a socket601 620
37500-37562
START_RANGE client19a socket602 630
37562-37625
START_RANGE client19a socket603 620
37625-37687
START_RANGE client19a socket604 630
37687-37750
START_RANGE client19a socket605 620
37750-37812
START_RANGE client19a socket606 630
37812-37875
START_RANGE client19a socket607 620
37875-37937
START_RANGE client19a socket608 630
37937-38000
START_RANGE client20a socket609 620
38000-38062
```

```
START_RANGE client20a socket610 630
38062-38125
START_RANGE client20a socket611 620
38125-38187
START_RANGE client20a socket612 630
38187-38250
START_RANGE client20a socket613 620
38250-38312
START_RANGE client20a socket614 630
38312-38375
START_RANGE client20a socket615 620
38375-38437
START_RANGE client20a socket616 630
38437-38500
START_RANGE client20a socket617 620
38500-38562
START_RANGE client20a socket618 630
38562-38625
START_RANGE client20a socket619 620
38625-38687
START_RANGE client20a socket620 630
38687-38750
START_RANGE client20a socket621 620
38750-38812
START_RANGE client20a socket622 630
38812-38875
START_RANGE client20a socket623 620
38875-38937
START_RANGE client20a socket624 630
38937-39000
START_RANGE client20a socket625 620
39000-39062
START_RANGE client20a socket626 630
39062-39125
START_RANGE client20a socket627 620
39125-39187
START_RANGE client20a socket628 630
39187-39250
START_RANGE client20a socket629 620
39250-39312
START_RANGE client20a socket630 630
39312-39375
START_RANGE client20a socket631 620
39375-39437
START_RANGE client20a socket632 630
39437-39500
START_RANGE client20a socket633 620
39500-39562
START_RANGE client20a socket634 630
39562-39625
START_RANGE client20a socket635 620
39625-39687
START_RANGE client20a socket636 630
39687-39750
START_RANGE client20a socket637 620
39750-39812
START_RANGE client20a socket638 630
39812-39875
START_RANGE client20a socket639 620
39875-39937
START_RANGE client20a socket640 630
39937-40000
#elif MASTER_NUM6
START_RANGE client21a socket641 620
40000-40062
START_RANGE client21a socket642 630
40062-40125
START_RANGE client21a socket643 620
40125-40187
START_RANGE client21a socket644 630
40187-40250
START_RANGE client21a socket645 620
40250-40312
START_RANGE client21a socket646 630
40312-40375
START_RANGE client21a socket647 620
40375-40437
START_RANGE client21a socket648 630
40437-40500
START_RANGE client21a socket649 620
40500-40562
START_RANGE client21a socket650 630
40562-40625
START_RANGE client21a socket651 620
40625-40687
START_RANGE client21a socket652 630
40687-40750
START_RANGE client21a socket653 620
40750-40812
START_RANGE client21a socket654 630
40812-40875
START_RANGE client21a socket655 620
40875-40937
START_RANGE client21a socket656 630
40937-41000
START_RANGE client21a socket657 620
41000-41062
START_RANGE client21a socket658 630
41062-41125
START_RANGE client21a socket659 620
41125-41187
START_RANGE client21a socket660 630
41187-41250
START_RANGE client21a socket661 620
41250-41312
START_RANGE client21a socket662 630
41312-41375
START_RANGE client21a socket663 620
41375-41437
START_RANGE client21a socket664 630
41437-41500
START_RANGE client21a socket665 620
41500-41562
START_RANGE client21a socket666 630
41562-41625
START_RANGE client21a socket667 620
41625-41687
START_RANGE client21a socket668 630
41687-41750
START_RANGE client21a socket669 620
41750-41812
START_RANGE client21a socket670 630
41812-41875
START_RANGE client21a socket671 620
41875-41937
START_RANGE client21a socket672 630
41937-42000
START_RANGE client22a socket673 620
42000-42062
START_RANGE client22a socket674 630
42062-42125
START_RANGE client22a socket675 620
42125-42187
START_RANGE client22a socket676 630
42187-42250
START_RANGE client22a socket677 620
42250-42312
START_RANGE client22a socket678 630
42312-42375
START_RANGE client22a socket679 620
42375-42437
START_RANGE client22a socket680 630
42437-42500
START_RANGE client22a socket681 620
42500-42562
START_RANGE client22a socket682 630
42562-42625
START_RANGE client22a socket683 620
42625-42687
START_RANGE client22a socket684 630
42687-42750
START_RANGE client22a socket685 620
42750-42812
START_RANGE client22a socket686 630
42812-42875
START_RANGE client22a socket687 620
42875-42937
START_RANGE client22a socket688 630
42937-43000
START_RANGE client22a socket689 620
43000-43062
START_RANGE client22a socket690 630
43062-43125
START_RANGE client22a socket691 620
43125-43187
START_RANGE client22a socket692 630
43187-43250
START_RANGE client22a socket693 620
43250-43312
START_RANGE client22a socket694 630
43312-43375
START_RANGE client22a socket695 620
43375-43437
START_RANGE client22a socket696 630
43437-43500
START_RANGE client22a socket697 620
43500-43562
START_RANGE client22a socket698 630
43562-43625
START_RANGE client22a socket699 620
43625-43687
START_RANGE client22a socket700 630
43687-43750
START_RANGE client22a socket701 620
43750-43812
START_RANGE client22a socket702 630
43812-43875
START_RANGE client22a socket703 620
43875-43937
START_RANGE client22a socket704 630
43937-44000
START_RANGE client23a socket705 620
44000-44062
START_RANGE client23a socket706 630
44062-44125
START_RANGE client23a socket707 620
44125-44187
START_RANGE client23a socket708 630
44187-44250
START_RANGE client23a socket709 620
44250-44312
START_RANGE client23a socket710 630
44312-44375
START_RANGE client23a socket711 620
44375-44437
START_RANGE client23a socket712 630
44437-44500
START_RANGE client23a socket713 620
44500-44562
START_RANGE client23a socket714 630
44562-44625
START_RANGE client23a socket715 620
44625-44687
START_RANGE client23a socket716 630
44687-44750
START_RANGE client23a socket717 620
44750-44812
START_RANGE client23a socket718 630
44812-44875
START_RANGE client23a socket719 620
44875-44937
START_RANGE client23a socket720 630
44937-45000
START_RANGE client23a socket721 620
45000-45062
START_RANGE client23a socket722 630
45062-45125
START_RANGE client23a socket723 620
45125-45187
START_RANGE client23a socket724 630
45187-45250
START_RANGE client23a socket725 620
45250-45312
START_RANGE client23a socket726 630
45312-45375
START_RANGE client23a socket727 620
45375-45437
START_RANGE client23a socket728 630
45437-45500
START_RANGE client23a socket729 620
45500-45562
START_RANGE client23a socket730 630
45562-45625
START_RANGE client23a socket731 620
45625-45687
START_RANGE client23a socket732 630
45687-45750
```

```
START_RANGE client23a socket733 620
45750-45812
START_RANGE client23a socket734 630
45812-45875
START_RANGE client23a socket735 620
45875-45937
START_RANGE client23a socket736 630
45937-46000
START_RANGE client24a socket737 620
46000-46062
START_RANGE client24a socket738 630
46062-46125
START_RANGE client24a socket739 620
46125-46187
START_RANGE client24a socket740 630
46187-46250
START_RANGE client24a socket741 620
46250-46312
START_RANGE client24a socket742 630
46312-46375
START_RANGE client24a socket743 620
46375-46437
START_RANGE client24a socket744 630
46437-46500
START_RANGE client24a socket745 620
46500-46562
START_RANGE client24a socket746 630
46562-46625
START_RANGE client24a socket747 620
46625-46687
START_RANGE client24a socket748 630
46687-46750
START_RANGE client24a socket749 620
46750-46812
START_RANGE client24a socket750 630
46812-46875
START_RANGE client24a socket751 620
46875-46937
START_RANGE client24a socket752 630
46937-47000
START_RANGE client24a socket753 620
47000-47062
START_RANGE client24a socket754 630
47062-47125
START_RANGE client24a socket755 620
47125-47187
START_RANGE client24a socket756 630
47187-47250
START_RANGE client24a socket757 620
47250-47312
START_RANGE client24a socket758 630
47312-47375
START_RANGE client24a socket759 620
47375-47437
START_RANGE client24a socket760 630
47437-47500
START_RANGE client24a socket761 620
47500-47562
START_RANGE client24a socket762 630
47562-47625
START_RANGE client24a socket763 620
47625-47687
START_RANGE client24a socket764 630
47687-47750
START_RANGE client24a socket765 620
47750-47812
START_RANGE client24a socket766 630
47812-47875
START_RANGE client24a socket767 620
47875-47937
START_RANGE client24a socket768 630
47937-48000
#elif MASTER_NUM7
START_RANGE client25a socket769 620
48000-48062
START_RANGE client25a socket770 630
48062-48125
START_RANGE client25a socket771 620
48125-48187
START_RANGE client25a socket772 630
48187-48250
START_RANGE client25a socket773 620
48250-48312
START_RANGE client25a socket774 630
48312-48375
START_RANGE client25a socket775 620
48375-48437
START_RANGE client25a socket776 630
48437-48500
START_RANGE client25a socket777 620
48500-48562
START_RANGE client25a socket778 630
48562-48625
START_RANGE client25a socket779 620
48625-48687
START_RANGE client25a socket780 630
48687-48750
START_RANGE client25a socket781 620
48750-48812
START_RANGE client25a socket782 630
48812-48875
START_RANGE client25a socket783 620
48875-48937
START_RANGE client25a socket784 630
48937-49000
START_RANGE client25a socket785 620
49000-49062
START_RANGE client25a socket786 630
49062-49125
START_RANGE client25a socket787 620
49125-49187
START_RANGE client25a socket788 630
49187-49250
START_RANGE client25a socket789 620
49250-49312
START_RANGE client25a socket790 630
49312-49375
START_RANGE client25a socket791 620
49375-49437
START_RANGE client25a socket792 630
49437-49500
START_RANGE client25a socket793 620
49500-49562
START_RANGE client25a socket794 630
49562-49625
START_RANGE client25a socket795 620
49625-49687
START_RANGE client25a socket796 630
49687-49750
START_RANGE client25a socket797 620
49750-49812
START_RANGE client25a socket798 630
49812-49875
START_RANGE client25a socket799 620
49875-49937
START_RANGE client25a socket800 630
49937-50000
START_RANGE client26a socket801 620
50000-50062
START_RANGE client26a socket802 630
50062-50125
START_RANGE client26a socket803 620
50125-50187
START_RANGE client26a socket804 630
50187-50250
START_RANGE client26a socket805 620
50250-50312
START_RANGE client26a socket806 630
50312-50375
START_RANGE client26a socket807 620
50375-50437
START_RANGE client26a socket808 630
50437-50500
START_RANGE client26a socket809 620
50500-50562
START_RANGE client26a socket810 630
50562-50625
START_RANGE client26a socket811 620
50625-50687
START_RANGE client26a socket812 630
50687-50750
START_RANGE client26a socket813 620
50750-50812
START_RANGE client26a socket814 630
50812-50875
START_RANGE client26a socket815 620
50875-50937
START_RANGE client26a socket816 630
50937-51000
START_RANGE client26a socket817 620
51000-51062
START_RANGE client26a socket818 630
51062-51125
START_RANGE client26a socket819 620
51125-51187
START_RANGE client26a socket820 630
51187-51250
START_RANGE client26a socket821 620
51250-51312
START_RANGE client26a socket822 630
51312-51375
START_RANGE client26a socket823 620
51375-51437
START_RANGE client26a socket824 630
51437-51500
START_RANGE client26a socket825 620
51500-51562
START_RANGE client26a socket826 630
51562-51625
START_RANGE client26a socket827 620
51625-51687
START_RANGE client26a socket828 630
51687-51750
START_RANGE client26a socket829 620
51750-51812
START_RANGE client26a socket830 630
51812-51875
START_RANGE client26a socket831 620
51875-51937
START_RANGE client26a socket832 630
51937-52000
START_RANGE client27a socket833 620
52000-52062
START_RANGE client27a socket834 630
52062-52125
START_RANGE client27a socket835 620
52125-52187
START_RANGE client27a socket836 630
52187-52250
START_RANGE client27a socket837 620
52250-52312
START_RANGE client27a socket838 630
52312-52375
START_RANGE client27a socket839 620
52375-52437
START_RANGE client27a socket840 630
52437-52500
START_RANGE client27a socket841 620
52500-52562
START_RANGE client27a socket842 630
52562-52625
START_RANGE client27a socket843 620
52625-52687
START_RANGE client27a socket844 630
52687-52750
START_RANGE client27a socket845 620
52750-52812
START_RANGE client27a socket846 630
52812-52875
START_RANGE client27a socket847 620
52875-52937
START_RANGE client27a socket848 630
52937-53000
START_RANGE client27a socket849 620
53000-53062
START_RANGE client27a socket850 630
53062-53125
START_RANGE client27a socket851 620
53125-53187
START_RANGE client27a socket852 630
53187-53250
START_RANGE client27a socket853 620
53250-53312
START_RANGE client27a socket854 630
53312-53375
START_RANGE client27a socket855 620
53375-53437
```

```
START_RANGE client27a socket856 630
53437-53500
START_RANGE client27a socket857 620
53500-53562
START_RANGE client27a socket858 630
53562-53625
START_RANGE client27a socket859 620
53625-53687
START_RANGE client27a socket860 630
53687-53750
START_RANGE client27a socket861 620
53750-53812
START_RANGE client27a socket862 630
53812-53875
START_RANGE client27a socket863 620
53875-53937
START_RANGE client27a socket864 630
53937-54000
START_RANGE client28a socket865 620
54000-54062
START_RANGE client28a socket866 630
54062-54125
START_RANGE client28a socket867 620
54125-54187
START_RANGE client28a socket868 630
54187-54250
START_RANGE client28a socket869 620
54250-54312
START_RANGE client28a socket870 630
54312-54375
START_RANGE client28a socket871 620
54375-54437
START_RANGE client28a socket872 630
54437-54500
START_RANGE client28a socket873 620
54500-54562
START_RANGE client28a socket874 630
54562-54625
START_RANGE client28a socket875 620
54625-54687
START_RANGE client28a socket876 630
54687-54750
START_RANGE client28a socket877 620
54750-54812
START_RANGE client28a socket878 630
54812-54875
START_RANGE client28a socket879 620
54875-54937
START_RANGE client28a socket880 630
54937-55000
START_RANGE client28a socket881 620
55000-55062
START_RANGE client28a socket882 630
55062-55125
START_RANGE client28a socket883 620
55125-55187
START_RANGE client28a socket884 630
55187-55250
START_RANGE client28a socket885 620
55250-55312
START_RANGE client28a socket886 630
55312-55375
START_RANGE client28a socket887 620
55375-55437
START_RANGE client28a socket888 630
55437-55500
START_RANGE client28a socket889 620
55500-55562
START_RANGE client28a socket890 630
55562-55625
START_RANGE client28a socket891 620
55625-55687
START_RANGE client28a socket892 630
55687-55750
START_RANGE client28a socket893 620
55750-55812
START_RANGE client28a socket894 630
55812-55875
START_RANGE client28a socket895 620
55875-55937
START_RANGE client28a socket896 630
55937-56000
#elif MASTER_NUM8
START_RANGE client29a socket897 620
56000-56062
START_RANGE client29a socket898 630
56062-56125
START_RANGE client29a socket899 620
56125-56187
START_RANGE client29a socket900 630
56187-56250
START_RANGE client29a socket901 620
56250-56312
START_RANGE client29a socket902 630
56312-56375
START_RANGE client29a socket903 620
56375-56437
START_RANGE client29a socket904 630
56437-56500
START_RANGE client29a socket905 620
56500-56562
START_RANGE client29a socket906 630
56562-56625
START_RANGE client29a socket907 620
56625-56687
START_RANGE client29a socket908 630
56687-56750
START_RANGE client29a socket909 620
56750-56812
START_RANGE client29a socket910 630
56812-56875
START_RANGE client29a socket911 620
56875-56937
START_RANGE client29a socket912 630
56937-57000
START_RANGE client29a socket913 620
57000-57062
START_RANGE client29a socket914 630
57062-57125
START_RANGE client29a socket915 620
57125-57187
START_RANGE client29a socket916 630
57187-57250
START_RANGE client29a socket917 620
57250-57312
START_RANGE client29a socket918 630
57312-57375
START_RANGE client29a socket919 620
57375-57437
START_RANGE client29a socket920 630
57437-57500
START_RANGE client29a socket921 620
57500-57562
START_RANGE client29a socket922 630
57562-57625
START_RANGE client29a socket923 620
57625-57687
START_RANGE client29a socket924 630
57687-57750
START_RANGE client29a socket925 620
57750-57812
START_RANGE client29a socket926 630
57812-57875
START_RANGE client29a socket927 620
57875-57937
START_RANGE client29a socket928 630
57937-58000
START_RANGE client30a socket929 620
58000-58062
START_RANGE client30a socket930 630
58062-58125
START_RANGE client30a socket931 620
58125-58187
START_RANGE client30a socket932 630
58187-58250
START_RANGE client30a socket933 620
58250-58312
START_RANGE client30a socket934 630
58312-58375
START_RANGE client30a socket935 620
58375-58437
START_RANGE client30a socket936 630
58437-58500
START_RANGE client30a socket937 620
58500-58562
START_RANGE client30a socket938 630
58562-58625
START_RANGE client30a socket939 620
58625-58687
START_RANGE client30a socket940 630
58687-58750
START_RANGE client30a socket941 620
58750-58812
START_RANGE client30a socket942 630
58812-58875
START_RANGE client30a socket943 620
58875-58937
START_RANGE client30a socket944 630
58937-59000
START_RANGE client30a socket945 620
59000-59062
START_RANGE client30a socket946 630
59062-59125
START_RANGE client30a socket947 620
59125-59187
START_RANGE client30a socket948 630
59187-59250
START_RANGE client30a socket949 620
59250-59312
START_RANGE client30a socket950 630
59312-59375
START_RANGE client30a socket951 620
59375-59437
START_RANGE client30a socket952 630
59437-59500
START_RANGE client30a socket953 620
59500-59562
START_RANGE client30a socket954 630
59562-59625
START_RANGE client30a socket955 620
59625-59687
START_RANGE client30a socket956 630
59687-59750
START_RANGE client30a socket957 620
59750-59812
START_RANGE client30a socket958 630
59812-59875
START_RANGE client30a socket959 620
59875-59937
START_RANGE client30a socket960 630
59937-60000
START_RANGE client31a socket961 620
60000-60062
START_RANGE client31a socket962 630
60062-60125
START_RANGE client31a socket963 620
60125-60187
START_RANGE client31a socket964 630
60187-60250
START_RANGE client31a socket965 620
60250-60312
START_RANGE client31a socket966 630
60312-60375
START_RANGE client31a socket967 620
60375-60437
START_RANGE client31a socket968 630
60437-60500
START_RANGE client31a socket969 620
60500-60562
START_RANGE client31a socket970 630
60562-60625
START_RANGE client31a socket971 620
60625-60687
START_RANGE client31a socket972 630
60687-60750
START_RANGE client31a socket973 620
60750-60812
START_RANGE client31a socket974 630
60812-60875
START_RANGE client31a socket975 620
60875-60937
START_RANGE client31a socket976 630
60937-61000
START_RANGE client31a socket977 620
61000-61062
START_RANGE client31a socket978 630
61062-61125
```

```
START_RANGE client31a socket979 620
61125-61187
START_RANGE client31a socket980 630
61187-61250
START_RANGE client31a socket981 620
61250-61312
START_RANGE client31a socket982 630
61312-61375
START_RANGE client31a socket983 620
61375-61437
START_RANGE client31a socket984 630
61437-61500
START_RANGE client31a socket985 620
61500-61562
START_RANGE client31a socket986 630
61562-61625
START_RANGE client31a socket987 620
61625-61687
START_RANGE client31a socket988 630
61687-61750
START_RANGE client31a socket989 620
61750-61812
START_RANGE client31a socket990 630
61812-61875
START_RANGE client31a socket991 620
61875-61937
START_RANGE client31a socket992 630
61937-62000
START_RANGE client32a socket993 620
62000-62062
START_RANGE client32a socket994 630
62062-62125
START_RANGE client32a socket995 620
62125-62187
START_RANGE client32a socket996 630
62187-62250
START_RANGE client32a socket997 620
62250-62312
START_RANGE client32a socket998 630
62312-62375
START_RANGE client32a socket999 620
62375-62437
START_RANGE client32a socket1000 630
62437-62500
START_RANGE client32a socket1001 620
62500-62562
START_RANGE client32a socket1002 630
62562-62625
START_RANGE client32a socket1003 620
62625-62687
START_RANGE client32a socket1004 630
62687-62750
START_RANGE client32a socket1005 620
62750-62812
START_RANGE client32a socket1006 630
62812-62875
START_RANGE client32a socket1007 620
62875-62937
START_RANGE client32a socket1008 630
62937-63000
START_RANGE client32a socket1009 620
63000-63062
START_RANGE client32a socket1010 630
63062-63125
START_RANGE client32a socket1011 620
63125-63187
START_RANGE client32a socket1012 630
63187-63250
START_RANGE client32a socket1013 620
63250-63312
START_RANGE client32a socket1014 630
63312-63375
START_RANGE client32a socket1015 620
63375-63437
START_RANGE client32a socket1016 630
63437-63500
START_RANGE client32a socket1017 620
63500-63562
START_RANGE client32a socket1018 630
63562-63625
START_RANGE client32a socket1019 620
63625-63687
START_RANGE client32a socket1020 630
63687-63750
START_RANGE client32a socket1021 620
63750-63812
START_RANGE client32a socket1022 630
63812-63875
START_RANGE client32a socket1023 620
63875-63937
START_RANGE client32a socket1024 630
63937-64000
#elif MASTER_NUM9
START_RANGE client33a socket1025 620
64000-64062
START_RANGE client33a socket1026 630
64062-64125
START_RANGE client33a socket1027 620
64125-64187
START_RANGE client33a socket1028 630
64187-64250
START_RANGE client33a socket1029 620
64250-64312
START_RANGE client33a socket1030 630
64312-64375
START_RANGE client33a socket1031 620
64375-64437
START_RANGE client33a socket1032 630
64437-64500
START_RANGE client33a socket1033 620
64500-64562
START_RANGE client33a socket1034 630
64562-64625
START_RANGE client33a socket1035 620
64625-64687
START_RANGE client33a socket1036 630
64687-64750
START_RANGE client33a socket1037 620
64750-64812
START_RANGE client33a socket1038 630
64812-64875
START_RANGE client33a socket1039 620
64875-64937
START_RANGE client33a socket1040 630
64937-65000
START_RANGE client33a socket1041 620
65000-65062
START_RANGE client33a socket1042 630
65062-65125
START_RANGE client33a socket1043 620
65125-65187
START_RANGE client33a socket1044 630
65187-65250
START_RANGE client33a socket1045 620
65250-65312
START_RANGE client33a socket1046 630
65312-65375
START_RANGE client33a socket1047 620
65375-65437
START_RANGE client33a socket1048 630
65437-65500
START_RANGE client33a socket1049 620
65500-65562
START_RANGE client33a socket1050 630
65562-65625
START_RANGE client33a socket1051 620
65625-65687
START_RANGE client33a socket1052 630
65687-65750
START_RANGE client33a socket1053 620
65750-65812
START_RANGE client33a socket1054 630
65812-65875
START_RANGE client33a socket1055 620
65875-65937
START_RANGE client33a socket1056 630
65937-66000
START_RANGE client34a socket1057 620
66000-66062
START_RANGE client34a socket1058 630
66062-66125
START_RANGE client34a socket1059 620
66125-66187
START_RANGE client34a socket1060 630
66187-66250
START_RANGE client34a socket1061 620
66250-66312
START_RANGE client34a socket1062 630
66312-66375
START_RANGE client34a socket1063 620
66375-66437
START_RANGE client34a socket1064 630
66437-66500
START_RANGE client34a socket1065 620
66500-66562
START_RANGE client34a socket1066 630
66562-66625
START_RANGE client34a socket1067 620
66625-66687
START_RANGE client34a socket1068 630
66687-66750
START_RANGE client34a socket1069 620
66750-66812
START_RANGE client34a socket1070 630
66812-66875
START_RANGE client34a socket1071 620
66875-66937
START_RANGE client34a socket1072 630
66937-67000
START_RANGE client34a socket1073 620
67000-67062
START_RANGE client34a socket1074 630
67062-67125
START_RANGE client34a socket1075 620
67125-67187
START_RANGE client34a socket1076 630
67187-67250
START_RANGE client34a socket1077 620
67250-67312
START_RANGE client34a socket1078 630
67312-67375
START_RANGE client34a socket1079 620
67375-67437
START_RANGE client34a socket1080 630
67437-67500
START_RANGE client34a socket1081 620
67500-67562
START_RANGE client34a socket1082 630
67562-67625
START_RANGE client34a socket1083 620
67625-67687
START_RANGE client34a socket1084 630
67687-67750
START_RANGE client34a socket1085 620
67750-67812
START_RANGE client34a socket1086 630
67812-67875
START_RANGE client34a socket1087 620
67875-67937
START_RANGE client34a socket1088 630
67937-68000
START_RANGE client35a socket1089 620
68000-68062
START_RANGE client35a socket1090 630
68062-68125
START_RANGE client35a socket1091 620
68125-68187
START_RANGE client35a socket1092 630
68187-68250
START_RANGE client35a socket1093 620
68250-68312
START_RANGE client35a socket1094 630
68312-68375
START_RANGE client35a socket1095 620
68375-68437
START_RANGE client35a socket1096 630
68437-68500
START_RANGE client35a socket1097 620
68500-68562
START_RANGE client35a socket1098 630
68562-68625
START_RANGE client35a socket1099 620
68625-68687
START_RANGE client35a socket1100 630
68687-68750
START_RANGE client35a socket1101 620
68750-68812
```

```
START_RANGE client35a socket1102 630
68812-68875
START_RANGE client35a socket1103 620
68875-68937
START_RANGE client35a socket1104 630
68937-69000
START_RANGE client35a socket1105 620
69000-69062
START_RANGE client35a socket1106 630
69062-69125
START_RANGE client35a socket1107 620
69125-69187
START_RANGE client35a socket1108 630
69187-69250
START_RANGE client35a socket1109 620
69250-69312
START_RANGE client35a socket1110 630
69312-69375
START_RANGE client35a socket1111 620
69375-69437
START_RANGE client35a socket1112 630
69437-69500
START_RANGE client35a socket1113 620
69500-69562
START_RANGE client35a socket1114 630
69562-69625
START_RANGE client35a socket1115 620
69625-69687
START_RANGE client35a socket1116 630
69687-69750
START_RANGE client35a socket1117 620
69750-69812
START_RANGE client35a socket1118 630
69812-69875
START_RANGE client35a socket1119 620
69875-69937
START_RANGE client35a socket1120 630
69937-70000
START_RANGE client36a socket1121 620
70000-70062
START_RANGE client36a socket1122 630
70062-70125
START_RANGE client36a socket1123 620
70125-70187
START_RANGE client36a socket1124 630
70187-70250
START_RANGE client36a socket1125 620
70250-70312
START_RANGE client36a socket1126 630
70312-70375
START_RANGE client36a socket1127 620
70375-70437
START_RANGE client36a socket1128 630
70437-70500
START_RANGE client36a socket1129 620
70500-70562
START_RANGE client36a socket1130 630
70562-70625
START_RANGE client36a socket1131 620
70625-70687
START_RANGE client36a socket1132 630
70687-70750
START_RANGE client36a socket1133 620
70750-70812
START_RANGE client36a socket1134 630
70812-70875
START_RANGE client36a socket1135 620
70875-70937
START_RANGE client36a socket1136 630
70937-71000
START_RANGE client36a socket1137 620
71000-71062
START_RANGE client36a socket1138 630
71062-71125
START_RANGE client36a socket1139 620
71125-71187
START_RANGE client36a socket1140 630
71187-71250
START_RANGE client36a socket1141 620
71250-71312
START_RANGE client36a socket1142 630
71312-71375
START_RANGE client36a socket1143 620
71375-71437
START_RANGE client36a socket1144 630
71437-71500
START_RANGE client36a socket1145 620
71500-71562
START_RANGE client36a socket1146 630
71562-71625
START_RANGE client36a socket1147 620
71625-71687
START_RANGE client36a socket1148 630
71687-71750
START_RANGE client36a socket1149 620
71750-71812
START_RANGE client36a socket1150 630
71812-71875
START_RANGE client36a socket1151 620
71875-71937
START_RANGE client36a socket1152 630
71937-72000
#elif MASTER_NUM10
START_RANGE client37a socket1153 620
72000-72062
START_RANGE client37a socket1154 630
72062-72125
START_RANGE client37a socket1155 620
72125-72187
START_RANGE client37a socket1156 630
72187-72250
START_RANGE client37a socket1157 620
72250-72312
START_RANGE client37a socket1158 630
72312-72375
START_RANGE client37a socket1159 620
72375-72437
START_RANGE client37a socket1160 630
72437-72500
START_RANGE client37a socket1161 620
72500-72562
START_RANGE client37a socket1162 630
72562-72625
START_RANGE client37a socket1163 620
72625-72687
START_RANGE client37a socket1164 630
72687-72750
START_RANGE client37a socket1165 620
72750-72812
START_RANGE client37a socket1166 630
72812-72875
START_RANGE client37a socket1167 620
72875-72937
START_RANGE client37a socket1168 630
72937-73000
START_RANGE client37a socket1169 620
73000-73062
START_RANGE client37a socket1170 630
73062-73125
START_RANGE client37a socket1171 620
73125-73187
START_RANGE client37a socket1172 630
73187-73250
START_RANGE client37a socket1173 620
73250-73312
START_RANGE client37a socket1174 630
73312-73375
START_RANGE client37a socket1175 620
73375-73437
START_RANGE client37a socket1176 630
73437-73500
START_RANGE client37a socket1177 620
73500-73562
START_RANGE client37a socket1178 630
73562-73625
START_RANGE client37a socket1179 620
73625-73687
START_RANGE client37a socket1180 630
73687-73750
START_RANGE client37a socket1181 620
73750-73812
START_RANGE client37a socket1182 630
73812-73875
START_RANGE client37a socket1183 620
73875-73937
START_RANGE client37a socket1184 630
73937-74000
START_RANGE client38a socket1185 620
74000-74062
START_RANGE client38a socket1186 630
74062-74125
START_RANGE client38a socket1187 620
74125-74187
START_RANGE client38a socket1188 630
74187-74250
START_RANGE client38a socket1189 620
74250-74312
START_RANGE client38a socket1190 630
74312-74375
START_RANGE client38a socket1191 620
74375-74437
START_RANGE client38a socket1192 630
74437-74500
START_RANGE client38a socket1193 620
74500-74562
START_RANGE client38a socket1194 630
74562-74625
START_RANGE client38a socket1195 620
74625-74687
START_RANGE client38a socket1196 630
74687-74750
START_RANGE client38a socket1197 620
74750-74812
START_RANGE client38a socket1198 630
74812-74875
START_RANGE client38a socket1199 620
74875-74937
START_RANGE client38a socket1200 630
74937-75000
START_RANGE client38a socket1201 620
75000-75062
START_RANGE client38a socket1202 630
75062-75125
START_RANGE client38a socket1203 620
75125-75187
START_RANGE client38a socket1204 630
75187-75250
START_RANGE client38a socket1205 620
75250-75312
START_RANGE client38a socket1206 630
75312-75375
START_RANGE client38a socket1207 620
75375-75437
START_RANGE client38a socket1208 630
75437-75500
START_RANGE client38a socket1209 620
75500-75562
START_RANGE client38a socket1210 630
75562-75625
START_RANGE client38a socket1211 620
75625-75687
START_RANGE client38a socket1212 630
75687-75750
START_RANGE client38a socket1213 620
75750-75812
START_RANGE client38a socket1214 630
75812-75875
START_RANGE client38a socket1215 620
75875-75937
START_RANGE client38a socket1216 630
75937-76000
START_RANGE client39a socket1217 620
76000-76062
START_RANGE client39a socket1218 630
76062-76125
START_RANGE client39a socket1219 620
76125-76187
START_RANGE client39a socket1220 630
76187-76250
START_RANGE client39a socket1221 620
76250-76312
START_RANGE client39a socket1222 630
76312-76375
START_RANGE client39a socket1223 620
76375-76437
START_RANGE client39a socket1224 630
76437-76500
```

```
START_RANGE client39a socket1225 620
76500-76562
START_RANGE client39a socket1226 630
76562-76625
START_RANGE client39a socket1227 620
76625-76687
START_RANGE client39a socket1228 630
76687-76750
START_RANGE client39a socket1229 620
76750-76812
START_RANGE client39a socket1230 630
76812-76875
START_RANGE client39a socket1231 620
76875-76937
START_RANGE client39a socket1232 630
76937-77000
START_RANGE client39a socket1233 620
77000-77062
START_RANGE client39a socket1234 630
77062-77125
START_RANGE client39a socket1235 620
77125-77187
START_RANGE client39a socket1236 630
77187-77250
START_RANGE client39a socket1237 620
77250-77312
START_RANGE client39a socket1238 630
77312-77375
START_RANGE client39a socket1239 620
77375-77437
START_RANGE client39a socket1240 630
77437-77500
START_RANGE client39a socket1241 620
77500-77562
START_RANGE client39a socket1242 630
77562-77625
START_RANGE client39a socket1243 620
77625-77687
START_RANGE client39a socket1244 630
77687-77750
START_RANGE client39a socket1245 620
77750-77812
START_RANGE client39a socket1246 630
77812-77875
START_RANGE client39a socket1247 620
77875-77937
START_RANGE client39a socket1248 630
77937-78000
START_RANGE client40a socket1249 620
78000-78062
START_RANGE client40a socket1250 630
78062-78125
START_RANGE client40a socket1251 620
78125-78187
START_RANGE client40a socket1252 630
78187-78250
START_RANGE client40a socket1253 620
78250-78312
START_RANGE client40a socket1254 630
78312-78375
START_RANGE client40a socket1255 620
78375-78437
START_RANGE client40a socket1256 630
78437-78500
START_RANGE client40a socket1257 620
78500-78562
START_RANGE client40a socket1258 630
78562-78625
START_RANGE client40a socket1259 620
78625-78687
START_RANGE client40a socket1260 630
78687-78750
START_RANGE client40a socket1261 620
78750-78812
START_RANGE client40a socket1262 630
78812-78875
START_RANGE client40a socket1263 620
78875-78937
START_RANGE client40a socket1264 630
78937-79000
START_RANGE client40a socket1265 620
79000-79062
START_RANGE client40a socket1266 630
79062-79125
START_RANGE client40a socket1267 620
79125-79187
START_RANGE client40a socket1268 630
79187-79250
START_RANGE client40a socket1269 620
79250-79312
START_RANGE client40a socket1270 630
79312-79375
START_RANGE client40a socket1271 620
79375-79437
START_RANGE client40a socket1272 630
79437-79500
START_RANGE client40a socket1273 620
79500-79562
START_RANGE client40a socket1274 630
79562-79625
START_RANGE client40a socket1275 620
79625-79687
START_RANGE client40a socket1276 630
79687-79750
START_RANGE client40a socket1277 620
79750-79812
START_RANGE client40a socket1278 630
79812-79875
START_RANGE client40a socket1279 620
79875-79937
START_RANGE client40a socket1280 630
79937-80000
#elif MASTER_NUM11
START_RANGE client41a socket1281 620
80000-80062
START_RANGE client41a socket1282 630
80062-80125
START_RANGE client41a socket1283 620
80125-80187
START_RANGE client41a socket1284 630
80187-80250
START_RANGE client41a socket1285 620
80250-80312
START_RANGE client41a socket1286 630
80312-80375
START_RANGE client41a socket1287 620
80375-80437
START_RANGE client41a socket1288 630
80437-80500
START_RANGE client41a socket1289 620
80500-80562
START_RANGE client41a socket1290 630
80562-80625
START_RANGE client41a socket1291 620
80625-80687
START_RANGE client41a socket1292 630
80687-80750
START_RANGE client41a socket1293 620
80750-80812
START_RANGE client41a socket1294 630
80812-80875
START_RANGE client41a socket1295 620
80875-80937
START_RANGE client41a socket1296 630
80937-81000
START_RANGE client41a socket1297 620
81000-81062
START_RANGE client41a socket1298 630
81062-81125
START_RANGE client41a socket1299 620
81125-81187
START_RANGE client41a socket1300 630
81187-81250
START_RANGE client41a socket1301 620
81250-81312
START_RANGE client41a socket1302 630
81312-81375
START_RANGE client41a socket1303 620
81375-81437
START_RANGE client41a socket1304 630
81437-81500
START_RANGE client41a socket1305 620
81500-81562
START_RANGE client41a socket1306 630
81562-81625
START_RANGE client41a socket1307 620
81625-81687
START_RANGE client41a socket1308 630
81687-81750
START_RANGE client41a socket1309 620
81750-81812
START_RANGE client41a socket1310 630
81812-81875
START_RANGE client41a socket1311 620
81875-81937
START_RANGE client41a socket1312 630
81937-82000
START_RANGE client42a socket1313 620
82000-82062
START_RANGE client42a socket1314 630
82062-82125
START_RANGE client42a socket1315 620
82125-82187
START_RANGE client42a socket1316 630
82187-82250
START_RANGE client42a socket1317 620
82250-82312
START_RANGE client42a socket1318 630
82312-82375
START_RANGE client42a socket1319 620
82375-82437
START_RANGE client42a socket1320 630
82437-82500
START_RANGE client42a socket1321 620
82500-82562
START_RANGE client42a socket1322 630
82562-82625
START_RANGE client42a socket1323 620
82625-82687
START_RANGE client42a socket1324 630
82687-82750
START_RANGE client42a socket1325 620
82750-82812
START_RANGE client42a socket1326 630
82812-82875
START_RANGE client42a socket1327 620
82875-82937
START_RANGE client42a socket1328 630
82937-83000
START_RANGE client42a socket1329 620
83000-83062
START_RANGE client42a socket1330 630
83062-83125
START_RANGE client42a socket1331 620
83125-83187
START_RANGE client42a socket1332 630
83187-83250
START_RANGE client42a socket1333 620
83250-83312
START_RANGE client42a socket1334 630
83312-83375
START_RANGE client42a socket1335 620
83375-83437
START_RANGE client42a socket1336 630
83437-83500
START_RANGE client42a socket1337 620
83500-83562
START_RANGE client42a socket1338 630
83562-83625
START_RANGE client42a socket1339 620
83625-83687
START_RANGE client42a socket1340 630
83687-83750
START_RANGE client42a socket1341 620
83750-83812
START_RANGE client42a socket1342 630
83812-83875
START_RANGE client42a socket1343 620
83875-83937
START_RANGE client42a socket1344 630
83937-84000
START_RANGE client43a socket1345 620
84000-84062
START_RANGE client43a socket1346 630
84062-84125
START_RANGE client43a socket1347 620
84125-84187
```

```
START_RANGE client43a socket1348 630
84187-84250
START_RANGE client43a socket1349 620
84250-84312
START_RANGE client43a socket1350 630
84312-84375
START_RANGE client43a socket1351 620
84375-84437
START_RANGE client43a socket1352 630
84437-84500
START_RANGE client43a socket1353 620
84500-84562
START_RANGE client43a socket1354 630
84562-84625
START_RANGE client43a socket1355 620
84625-84687
START_RANGE client43a socket1356 630
84687-84750
START_RANGE client43a socket1357 620
84750-84812
START_RANGE client43a socket1358 630
84812-84875
START_RANGE client43a socket1359 620
84875-84937
START_RANGE client43a socket1360 630
84937-85000
START_RANGE client43a socket1361 620
85000-85062
START_RANGE client43a socket1362 630
85062-85125
START_RANGE client43a socket1363 620
85125-85187
START_RANGE client43a socket1364 630
85187-85250
START_RANGE client43a socket1365 620
85250-85312
START_RANGE client43a socket1366 630
85312-85375
START_RANGE client43a socket1367 620
85375-85437
START_RANGE client43a socket1368 630
85437-85500
START_RANGE client43a socket1369 620
85500-85562
START_RANGE client43a socket1370 630
85562-85625
START_RANGE client43a socket1371 620
85625-85687
START_RANGE client43a socket1372 630
85687-85750
START_RANGE client43a socket1373 620
85750-85812
START_RANGE client43a socket1374 630
85812-85875
START_RANGE client43a socket1375 620
85875-85937
START_RANGE client43a socket1376 630
85937-86000
START_RANGE client44a socket1377 620
86000-86062
START_RANGE client44a socket1378 630
86062-86125
START_RANGE client44a socket1379 620
86125-86187
START_RANGE client44a socket1380 630
86187-86250
START_RANGE client44a socket1381 620
86250-86312
START_RANGE client44a socket1382 630
86312-86375
START_RANGE client44a socket1383 620
86375-86437
START_RANGE client44a socket1384 630
86437-86500
START_RANGE client44a socket1385 620
86500-86562
START_RANGE client44a socket1386 630
86562-86625
START_RANGE client44a socket1387 620
86625-86687
START_RANGE client44a socket1388 630
86687-86750
START_RANGE client44a socket1389 620
86750-86812
START_RANGE client44a socket1390 630
86812-86875
START_RANGE client44a socket1391 620
86875-86937
START_RANGE client44a socket1392 630
86937-87000
START_RANGE client44a socket1393 620
87000-87062
START_RANGE client44a socket1394 630
87062-87125
START_RANGE client44a socket1395 620
87125-87187
START_RANGE client44a socket1396 630
87187-87250
START_RANGE client44a socket1397 620
87250-87312
START_RANGE client44a socket1398 630
87312-87375
START_RANGE client44a socket1399 620
87375-87437
START_RANGE client44a socket1400 630
87437-87500
START_RANGE client44a socket1401 620
87500-87562
START_RANGE client44a socket1402 630
87562-87625
START_RANGE client44a socket1403 620
87625-87687
START_RANGE client44a socket1404 630
87687-87750
START_RANGE client44a socket1405 620
87750-87812
START_RANGE client44a socket1406 630
87812-87875
START_RANGE client44a socket1407 620
87875-87937
START_RANGE client44a socket1408 630
87937-88000
#elif MASTER_NUM12
START_RANGE client45a socket1409 620
88000-88062
START_RANGE client45a socket1410 630
88062-88125
START_RANGE client45a socket1411 620
88125-88187
START_RANGE client45a socket1412 630
88187-88250
START_RANGE client45a socket1413 620
88250-88312
START_RANGE client45a socket1414 630
88312-88375
START_RANGE client45a socket1415 620
88375-88437
START_RANGE client45a socket1416 630
88437-88500
START_RANGE client45a socket1417 620
88500-88562
START_RANGE client45a socket1418 630
88562-88625
START_RANGE client45a socket1419 620
88625-88687
START_RANGE client45a socket1420 630
88687-88750
START_RANGE client45a socket1421 620
88750-88812
START_RANGE client45a socket1422 630
88812-88875
START_RANGE client45a socket1423 620
88875-88937
START_RANGE client45a socket1424 630
88937-89000
START_RANGE client45a socket1425 620
89000-89062
START_RANGE client45a socket1426 630
89062-89125
START_RANGE client45a socket1427 620
89125-89187
START_RANGE client45a socket1428 630
89187-89250
START_RANGE client45a socket1429 620
89250-89312
START_RANGE client45a socket1430 630
89312-89375
START_RANGE client45a socket1431 620
89375-89437
START_RANGE client45a socket1432 630
89437-89500
START_RANGE client45a socket1433 620
89500-89562
START_RANGE client45a socket1434 630
89562-89625
START_RANGE client45a socket1435 620
89625-89687
START_RANGE client45a socket1436 630
89687-89750
START_RANGE client45a socket1437 620
89750-89812
START_RANGE client45a socket1438 630
89812-89875
START_RANGE client45a socket1439 620
89875-89937
START_RANGE client45a socket1440 630
89937-90000
START_RANGE client46a socket1441 620
90000-90062
START_RANGE client46a socket1442 630
90062-90125
START_RANGE client46a socket1443 620
90125-90187
START_RANGE client46a socket1444 630
90187-90250
START_RANGE client46a socket1445 620
90250-90312
START_RANGE client46a socket1446 630
90312-90375
START_RANGE client46a socket1447 620
90375-90437
START_RANGE client46a socket1448 630
90437-90500
START_RANGE client46a socket1449 620
90500-90562
START_RANGE client46a socket1450 630
90562-90625
START_RANGE client46a socket1451 620
90625-90687
START_RANGE client46a socket1452 630
90687-90750
START_RANGE client46a socket1453 620
90750-90812
START_RANGE client46a socket1454 630
90812-90875
START_RANGE client46a socket1455 620
90875-90937
START_RANGE client46a socket1456 630
90937-91000
START_RANGE client46a socket1457 620
91000-91062
START_RANGE client46a socket1458 630
91062-91125
START_RANGE client46a socket1459 620
91125-91187
START_RANGE client46a socket1460 630
91187-91250
START_RANGE client46a socket1461 620
91250-91312
START_RANGE client46a socket1462 630
91312-91375
START_RANGE client46a socket1463 620
91375-91437
START_RANGE client46a socket1464 630
91437-91500
START_RANGE client46a socket1465 620
91500-91562
START_RANGE client46a socket1466 630
91562-91625
START_RANGE client46a socket1467 620
91625-91687
START_RANGE client46a socket1468 630
91687-91750
START_RANGE client46a socket1469 620
91750-91812
START_RANGE client46a socket1470 630
91812-91875
```

```
START_RANGE client46a socket1471 620
91875-91937
START_RANGE client46a socket1472 630
91937-92000
START_RANGE client47a socket1473 620
92000-92062
START_RANGE client47a socket1474 630
92062-92125
START_RANGE client47a socket1475 620
92125-92187
START_RANGE client47a socket1476 630
92187-92250
START_RANGE client47a socket1477 620
92250-92312
START_RANGE client47a socket1478 630
92312-92375
START_RANGE client47a socket1479 620
92375-92437
START_RANGE client47a socket1480 630
92437-92500
START_RANGE client47a socket1481 620
92500-92562
START_RANGE client47a socket1482 630
92562-92625
START_RANGE client47a socket1483 620
92625-92687
START_RANGE client47a socket1484 630
92687-92750
START_RANGE client47a socket1485 620
92750-92812
START_RANGE client47a socket1486 630
92812-92875
START_RANGE client47a socket1487 620
92875-92937
START_RANGE client47a socket1488 630
92937-93000
START_RANGE client47a socket1489 620
93000-93062
START_RANGE client47a socket1490 630
93062-93125
START_RANGE client47a socket1491 620
93125-93187
START_RANGE client47a socket1492 630
93187-93250
START_RANGE client47a socket1493 620
93250-93312
START_RANGE client47a socket1494 630
93312-93375
START_RANGE client47a socket1495 620
93375-93437
START_RANGE client47a socket1496 630
93437-93500
START_RANGE client47a socket1497 620
93500-93562
START_RANGE client47a socket1498 630
93562-93625
START_RANGE client47a socket1499 620
93625-93687
START_RANGE client47a socket1500 630
93687-93750
START_RANGE client47a socket1501 620
93750-93812
START_RANGE client47a socket1502 630
93812-93875
START_RANGE client47a socket1503 620
93875-93937
START_RANGE client47a socket1504 630
93937-94000
START_RANGE client48a socket1505 620
94000-94062
START_RANGE client48a socket1506 630
94062-94125
START_RANGE client48a socket1507 620
94125-94187
START_RANGE client48a socket1508 630
94187-94250
START_RANGE client48a socket1509 620
94250-94312
START_RANGE client48a socket1510 630
94312-94375
START_RANGE client48a socket1511 620
94375-94437
START_RANGE client48a socket1512 630
94437-94500
START_RANGE client48a socket1513 620
94500-94562
START_RANGE client48a socket1514 630
94562-94625
START_RANGE client48a socket1515 620
94625-94687
START_RANGE client48a socket1516 630
94687-94750
START_RANGE client48a socket1517 620
94750-94812
START_RANGE client48a socket1518 630
94812-94875
START_RANGE client48a socket1519 620
94875-94937
START_RANGE client48a socket1520 630
94937-95000
START_RANGE client48a socket1521 620
95000-95062
START_RANGE client48a socket1522 630
95062-95125
START_RANGE client48a socket1523 620
95125-95187
START_RANGE client48a socket1524 630
95187-95250
START_RANGE client48a socket1525 620
95250-95312
START_RANGE client48a socket1526 630
95312-95375
START_RANGE client48a socket1527 620
95375-95437
START_RANGE client48a socket1528 630
95437-95500
START_RANGE client48a socket1529 620
95500-95562
START_RANGE client48a socket1530 630
95562-95625
START_RANGE client48a socket1531 620
95625-95687
START_RANGE client48a socket1532 630
95687-95750
START_RANGE client48a socket1533 620
95750-95812
START_RANGE client48a socket1534 630
95812-95875
START_RANGE client48a socket1535 620
95875-95937
START_RANGE client48a socket1536 630
95937-96000
#elif MASTER_NUM13
START_RANGE client49a socket1537 620
96000-96062
START_RANGE client49a socket1538 630
96062-96125
START_RANGE client49a socket1539 620
96125-96187
START_RANGE client49a socket1540 630
96187-96250
START_RANGE client49a socket1541 620
96250-96312
START_RANGE client49a socket1542 630
96312-96375
START_RANGE client49a socket1543 620
96375-96437
START_RANGE client49a socket1544 630
96437-96500
START_RANGE client49a socket1545 620
96500-96562
START_RANGE client49a socket1546 630
96562-96625
START_RANGE client49a socket1547 620
96625-96687
START_RANGE client49a socket1548 630
96687-96750
START_RANGE client49a socket1549 620
96750-96812
START_RANGE client49a socket1550 630
96812-96875
START_RANGE client49a socket1551 620
96875-96937
START_RANGE client49a socket1552 630
96937-97000
START_RANGE client49a socket1553 620
97000-97062
START_RANGE client49a socket1554 630
97062-97125
START_RANGE client49a socket1555 620
97125-97187
START_RANGE client49a socket1556 630
97187-97250
START_RANGE client49a socket1557 620
97250-97312
START_RANGE client49a socket1558 630
97312-97375
START_RANGE client49a socket1559 620
97375-97437
START_RANGE client49a socket1560 630
97437-97500
START_RANGE client49a socket1561 620
97500-97562
START_RANGE client49a socket1562 630
97562-97625
START_RANGE client49a socket1563 620
97625-97687
START_RANGE client49a socket1564 630
97687-97750
START_RANGE client49a socket1565 620
97750-97812
START_RANGE client49a socket1566 630
97812-97875
START_RANGE client49a socket1567 620
97875-97937
START_RANGE client49a socket1568 630
97937-98000
START_RANGE client50a socket1569 620
98000-98062
START_RANGE client50a socket1570 630
98062-98125
START_RANGE client50a socket1571 620
98125-98187
START_RANGE client50a socket1572 630
98187-98250
START_RANGE client50a socket1573 620
98250-98312
START_RANGE client50a socket1574 630
98312-98375
START_RANGE client50a socket1575 620
98375-98437
START_RANGE client50a socket1576 630
98437-98500
START_RANGE client50a socket1577 620
98500-98562
START_RANGE client50a socket1578 630
98562-98625
START_RANGE client50a socket1579 620
98625-98687
START_RANGE client50a socket1580 630
98687-98750
START_RANGE client50a socket1581 620
98750-98812
START_RANGE client50a socket1582 630
98812-98875
START_RANGE client50a socket1583 620
98875-98937
START_RANGE client50a socket1584 630
98937-99000
START_RANGE client50a socket1585 620
99000-99062
START_RANGE client50a socket1586 630
99062-99125
START_RANGE client50a socket1587 620
99125-99187
START_RANGE client50a socket1588 630
99187-99250
START_RANGE client50a socket1589 620
99250-99312
START_RANGE client50a socket1590 630
99312-99375
START_RANGE client50a socket1591 620
99375-99437
START_RANGE client50a socket1592 630
99437-99500
START_RANGE client50a socket1593 620
99500-99562
```

```
START_RANGE client50a socket1594 630
99562-99625
START_RANGE client50a socket1595 620
99625-99687
START_RANGE client50a socket1596 630
99687-99750
START_RANGE client50a socket1597 620
99750-99812
START_RANGE client50a socket1598 630
99812-99875
START_RANGE client50a socket1599 620
99875-99937
START_RANGE client50a socket1600 630
99937-100000
START_RANGE client51a socket1601 620
100000-100062
START_RANGE client51a socket1602 630
100062-100125
START_RANGE client51a socket1603 620
100125-100187
START_RANGE client51a socket1604 630
100187-100250
START_RANGE client51a socket1605 620
100250-100312
START_RANGE client51a socket1606 630
100312-100375
START_RANGE client51a socket1607 620
100375-100437
START_RANGE client51a socket1608 630
100437-100500
START_RANGE client51a socket1609 620
100500-100562
START_RANGE client51a socket1610 630
100562-100625
START_RANGE client51a socket1611 620
100625-100687
START_RANGE client51a socket1612 630
100687-100750
START_RANGE client51a socket1613 620
100750-100812
START_RANGE client51a socket1614 630
100812-100875
START_RANGE client51a socket1615 620
100875-100937
START_RANGE client51a socket1616 630
100937-101000
START_RANGE client51a socket1617 620
101000-101062
START_RANGE client51a socket1618 630
101062-101125
START_RANGE client51a socket1619 620
101125-101187
START_RANGE client51a socket1620 630
101187-101250
START_RANGE client51a socket1621 620
101250-101312
START_RANGE client51a socket1622 630
101312-101375
START_RANGE client51a socket1623 620
101375-101437
START_RANGE client51a socket1624 630
101437-101500
START_RANGE client51a socket1625 620
101500-101562
START_RANGE client51a socket1626 630
101562-101625
START_RANGE client51a socket1627 620
101625-101687
START_RANGE client51a socket1628 630
101687-101750
START_RANGE client51a socket1629 620
101750-101812
START_RANGE client51a socket1630 630
101812-101875
START_RANGE client51a socket1631 620
101875-101937
START_RANGE client51a socket1632 630
101937-102000
START_RANGE client52a socket1633 620
102000-102062
START_RANGE client52a socket1634 630
102062-102125
START_RANGE client52a socket1635 620
102125-102187
START_RANGE client52a socket1636 630
102187-102250
START_RANGE client52a socket1637 620
102250-102312
START_RANGE client52a socket1638 630
102312-102375
START_RANGE client52a socket1639 620
102375-102437
START_RANGE client52a socket1640 630
102437-102500
START_RANGE client52a socket1641 620
102500-102562
START_RANGE client52a socket1642 630
102562-102625
START_RANGE client52a socket1643 620
102625-102687
START_RANGE client52a socket1644 630
102687-102750
START_RANGE client52a socket1645 620
102750-102812
START_RANGE client52a socket1646 630
102812-102875
START_RANGE client52a socket1647 620
102875-102937
START_RANGE client52a socket1648 630
102937-103000
START_RANGE client52a socket1649 620
103000-103062
START_RANGE client52a socket1650 630
103062-103125
START_RANGE client52a socket1651 620
103125-103187
START_RANGE client52a socket1652 630
103187-103250
START_RANGE client52a socket1653 620
103250-103312
START_RANGE client52a socket1654 630
103312-103375
START_RANGE client52a socket1655 620
103375-103437
START_RANGE client52a socket1656 630
103437-103500
START_RANGE client52a socket1657 620
103500-103562
START_RANGE client52a socket1658 630
103562-103625
START_RANGE client52a socket1659 620
103625-103687
START_RANGE client52a socket1660 630
103687-103750
START_RANGE client52a socket1661 620
103750-103812
START_RANGE client52a socket1662 630
103812-103875
START_RANGE client52a socket1663 620
103875-103937
START_RANGE client52a socket1664 630
103937-104000
#elif MASTER_NUM14
START_RANGE client53a socket1665 620
104000-104062
START_RANGE client53a socket1666 630
104062-104125
START_RANGE client53a socket1667 620
104125-104187
START_RANGE client53a socket1668 630
104187-104250
START_RANGE client53a socket1669 620
104250-104312
START_RANGE client53a socket1670 630
104312-104375
START_RANGE client53a socket1671 620
104375-104437
START_RANGE client53a socket1672 630
104437-104500
START_RANGE client53a socket1673 620
104500-104562
START_RANGE client53a socket1674 630
104562-104625
START_RANGE client53a socket1675 620
104625-104687
START_RANGE client53a socket1676 630
104687-104750
START_RANGE client53a socket1677 620
104750-104812
START_RANGE client53a socket1678 630
104812-104875
START_RANGE client53a socket1679 620
104875-104937
START_RANGE client53a socket1680 630
104937-105000
START_RANGE client53a socket1681 620
105000-105062
START_RANGE client53a socket1682 630
105062-105125
START_RANGE client53a socket1683 620
105125-105187
START_RANGE client53a socket1684 630
105187-105250
START_RANGE client53a socket1685 620
105250-105312
START_RANGE client53a socket1686 630
105312-105375
START_RANGE client53a socket1687 620
105375-105437
START_RANGE client53a socket1688 630
105437-105500
START_RANGE client53a socket1689 620
105500-105562
START_RANGE client53a socket1690 630
105562-105625
START_RANGE client53a socket1691 620
105625-105687
START_RANGE client53a socket1692 630
105687-105750
START_RANGE client53a socket1693 620
105750-105812
START_RANGE client53a socket1694 630
105812-105875
START_RANGE client53a socket1695 620
105875-105937
START_RANGE client53a socket1696 630
105937-106000
START_RANGE client54a socket1697 620
106000-106062
START_RANGE client54a socket1698 630
106062-106125
START_RANGE client54a socket1699 620
106125-106187
START_RANGE client54a socket1700 630
106187-106250
START_RANGE client54a socket1701 620
106250-106312
START_RANGE client54a socket1702 630
106312-106375
START_RANGE client54a socket1703 620
106375-106437
START_RANGE client54a socket1704 630
106437-106500
START_RANGE client54a socket1705 620
106500-106562
START_RANGE client54a socket1706 630
106562-106625
START_RANGE client54a socket1707 620
106625-106687
START_RANGE client54a socket1708 630
106687-106750
START_RANGE client54a socket1709 620
106750-106812
START_RANGE client54a socket1710 630
106812-106875
START_RANGE client54a socket1711 620
106875-106937
START_RANGE client54a socket1712 630
106937-107000
START_RANGE client54a socket1713 620
107000-107062
START_RANGE client54a socket1714 630
107062-107125
START_RANGE client54a socket1715 620
107125-107187
START_RANGE client54a socket1716 630
107187-107250
```

```
START_RANGE client54a socket1717 620
107250-107312
START_RANGE client54a socket1718 630
107312-107375
START_RANGE client54a socket1719 620
107375-107437
START_RANGE client54a socket1720 630
107437-107500
START_RANGE client54a socket1721 620
107500-107562
START_RANGE client54a socket1722 630
107562-107625
START_RANGE client54a socket1723 620
107625-107687
START_RANGE client54a socket1724 630
107687-107750
START_RANGE client54a socket1725 620
107750-107812
START_RANGE client54a socket1726 630
107812-107875
START_RANGE client54a socket1727 620
107875-107937
START_RANGE client54a socket1728 630
107937-108000
START_RANGE client55a socket1729 620
108000-108062
START_RANGE client55a socket1730 630
108062-108125
START_RANGE client55a socket1731 620
108125-108187
START_RANGE client55a socket1732 630
108187-108250
START_RANGE client55a socket1733 620
108250-108312
START_RANGE client55a socket1734 630
108312-108375
START_RANGE client55a socket1735 620
108375-108437
START_RANGE client55a socket1736 630
108437-108500
START_RANGE client55a socket1737 620
108500-108562
START_RANGE client55a socket1738 630
108562-108625
START_RANGE client55a socket1739 620
108625-108687
START_RANGE client55a socket1740 630
108687-108750
START_RANGE client55a socket1741 620
108750-108812
START_RANGE client55a socket1742 630
108812-108875
START_RANGE client55a socket1743 620
108875-108937
START_RANGE client55a socket1744 630
108937-109000
START_RANGE client55a socket1745 620
109000-109062
START_RANGE client55a socket1746 630
109062-109125
START_RANGE client55a socket1747 620
109125-109187
START_RANGE client55a socket1748 630
109187-109250
START_RANGE client55a socket1749 620
109250-109312
START_RANGE client55a socket1750 630
109312-109375
START_RANGE client55a socket1751 620
109375-109437
START_RANGE client55a socket1752 630
109437-109500
START_RANGE client55a socket1753 620
109500-109562
START_RANGE client55a socket1754 630
109562-109625
START_RANGE client55a socket1755 620
109625-109687
START_RANGE client55a socket1756 630
109687-109750
START_RANGE client55a socket1757 620
109750-109812
START_RANGE client55a socket1758 630
109812-109875
START_RANGE client55a socket1759 620
109875-109937
START_RANGE client55a socket1760 630
109937-110000
START_RANGE client56a socket1761 620
110000-110062
START_RANGE client56a socket1762 630
110062-110125
START_RANGE client56a socket1763 620
110125-110187
START_RANGE client56a socket1764 630
110187-110250
START_RANGE client56a socket1765 620
110250-110312
START_RANGE client56a socket1766 630
110312-110375
START_RANGE client56a socket1767 620
110375-110437
START_RANGE client56a socket1768 630
110437-110500
START_RANGE client56a socket1769 620
110500-110562
START_RANGE client56a socket1770 630
110562-110625
START_RANGE client56a socket1771 620
110625-110687
START_RANGE client56a socket1772 630
110687-110750
START_RANGE client56a socket1773 620
110750-110812
START_RANGE client56a socket1774 630
110812-110875
START_RANGE client56a socket1775 620
110875-110937
START_RANGE client56a socket1776 630
110937-111000
START_RANGE client56a socket1777 620
111000-111062
START_RANGE client56a socket1778 630
111062-111125
START_RANGE client56a socket1779 620
111125-111187
START_RANGE client56a socket1780 630
111187-111250
START_RANGE client56a socket1781 620
111250-111312
START_RANGE client56a socket1782 630
111312-111375
START_RANGE client56a socket1783 620
111375-111437
START_RANGE client56a socket1784 630
111437-111500
START_RANGE client56a socket1785 620
111500-111562
START_RANGE client56a socket1786 630
111562-111625
START_RANGE client56a socket1787 620
111625-111687
START_RANGE client56a socket1788 630
111687-111750
START_RANGE client56a socket1789 620
111750-111812
START_RANGE client56a socket1790 630
111812-111875
START_RANGE client56a socket1791 620
111875-111937
START_RANGE client56a socket1792 630
111937-112000
#elif MASTER_NUM15
START_RANGE client57a socket1793 620
112000-112062
START_RANGE client57a socket1794 630
112062-112125
START_RANGE client57a socket1795 620
112125-112187
START_RANGE client57a socket1796 630
112187-112250
START_RANGE client57a socket1797 620
112250-112312
START_RANGE client57a socket1798 630
112312-112375
START_RANGE client57a socket1799 620
112375-112437
START_RANGE client57a socket1800 630
112437-112500
START_RANGE client57a socket1801 620
112500-112562
START_RANGE client57a socket1802 630
112562-112625
START_RANGE client57a socket1803 620
112625-112687
START_RANGE client57a socket1804 630
112687-112750
START_RANGE client57a socket1805 620
112750-112812
START_RANGE client57a socket1806 630
112812-112875
START_RANGE client57a socket1807 620
112875-112937
START_RANGE client57a socket1808 630
112937-113000
START_RANGE client57a socket1809 620
113000-113062
START_RANGE client57a socket1810 630
113062-113125
START_RANGE client57a socket1811 620
113125-113187
START_RANGE client57a socket1812 630
113187-113250
START_RANGE client57a socket1813 620
113250-113312
START_RANGE client57a socket1814 630
113312-113375
START_RANGE client57a socket1815 620
113375-113437
START_RANGE client57a socket1816 630
113437-113500
START_RANGE client57a socket1817 620
113500-113562
START_RANGE client57a socket1818 630
113562-113625
START_RANGE client57a socket1819 620
113625-113687
START_RANGE client57a socket1820 630
113687-113750
START_RANGE client57a socket1821 620
113750-113812
START_RANGE client57a socket1822 630
113812-113875
START_RANGE client57a socket1823 620
113875-113937
START_RANGE client57a socket1824 630
113937-114000
START_RANGE client58a socket1825 620
114000-114062
START_RANGE client58a socket1826 630
114062-114125
START_RANGE client58a socket1827 620
114125-114187
START_RANGE client58a socket1828 630
114187-114250
START_RANGE client58a socket1829 620
114250-114312
START_RANGE client58a socket1830 630
114312-114375
START_RANGE client58a socket1831 620
114375-114437
START_RANGE client58a socket1832 630
114437-114500
START_RANGE client58a socket1833 620
114500-114562
START_RANGE client58a socket1834 630
114562-114625
START_RANGE client58a socket1835 620
114625-114687
START_RANGE client58a socket1836 630
114687-114750
START_RANGE client58a socket1837 620
114750-114812
START_RANGE client58a socket1838 630
114812-114875
START_RANGE client58a socket1839 620
114875-114937
```

```
START_RANGE client58a socket1840 630
114937-115000
START_RANGE client58a socket1841 620
115000-115062
START_RANGE client58a socket1842 630
115062-115125
START_RANGE client58a socket1843 620
115125-115187
START_RANGE client58a socket1844 630
115187-115250
START_RANGE client58a socket1845 620
115250-115312
START_RANGE client58a socket1846 630
115312-115375
START_RANGE client58a socket1847 620
115375-115437
START_RANGE client58a socket1848 630
115437-115500
START_RANGE client58a socket1849 620
115500-115562
START_RANGE client58a socket1850 630
115562-115625
START_RANGE client58a socket1851 620
115625-115687
START_RANGE client58a socket1852 630
115687-115750
START_RANGE client58a socket1853 620
115750-115812
START_RANGE client58a socket1854 630
115812-115875
START_RANGE client58a socket1855 620
115875-115937
START_RANGE client58a socket1856 630
115937-116000
START_RANGE client59a socket1857 620
116000-116062
START_RANGE client59a socket1858 630
116062-116125
START_RANGE client59a socket1859 620
116125-116187
START_RANGE client59a socket1860 630
116187-116250
START_RANGE client59a socket1861 620
116250-116312
START_RANGE client59a socket1862 630
116312-116375
START_RANGE client59a socket1863 620
116375-116437
START_RANGE client59a socket1864 630
116437-116500
START_RANGE client59a socket1865 620
116500-116562
START_RANGE client59a socket1866 630
116562-116625
START_RANGE client59a socket1867 620
116625-116687
START_RANGE client59a socket1868 630
116687-116750
START_RANGE client59a socket1869 620
116750-116812
START_RANGE client59a socket1870 630
116812-116875
START_RANGE client59a socket1871 620
116875-116937
START_RANGE client59a socket1872 630
116937-117000
START_RANGE client59a socket1873 620
117000-117062
START_RANGE client59a socket1874 630
117062-117125
START_RANGE client59a socket1875 620
117125-117187
START_RANGE client59a socket1876 630
117187-117250
START_RANGE client59a socket1877 620
117250-117312
START_RANGE client59a socket1878 630
117312-117375
START_RANGE client59a socket1879 620
117375-117437
START_RANGE client59a socket1880 630
117437-117500
START_RANGE client59a socket1881 620
117500-117562
START_RANGE client59a socket1882 630
117562-117625
START_RANGE client59a socket1883 620
117625-117687
START_RANGE client59a socket1884 630
117687-117750
START_RANGE client59a socket1885 620
117750-117812
START_RANGE client59a socket1886 630
117812-117875
START_RANGE client59a socket1887 620
117875-117937
START_RANGE client59a socket1888 630
117937-118000
START_RANGE client60a socket1889 620
118000-118062
START_RANGE client60a socket1890 630
118062-118125
START_RANGE client60a socket1891 620
118125-118187
START_RANGE client60a socket1892 630
118187-118250
START_RANGE client60a socket1893 620
118250-118312
START_RANGE client60a socket1894 630
118312-118375
START_RANGE client60a socket1895 620
118375-118437
START_RANGE client60a socket1896 630
118437-118500
START_RANGE client60a socket1897 620
118500-118562
START_RANGE client60a socket1898 630
118562-118625
START_RANGE client60a socket1899 620
118625-118687
START_RANGE client60a socket1900 630
118687-118750
START_RANGE client60a socket1901 620
118750-118812
START_RANGE client60a socket1902 630
118812-118875
START_RANGE client60a socket1903 620
118875-118937
START_RANGE client60a socket1904 630
118937-119000
START_RANGE client60a socket1905 620
119000-119062
START_RANGE client60a socket1906 630
119062-119125
START_RANGE client60a socket1907 620
119125-119187
START_RANGE client60a socket1908 630
119187-119250
START_RANGE client60a socket1909 620
119250-119312
START_RANGE client60a socket1910 630
119312-119375
START_RANGE client60a socket1911 620
119375-119437
START_RANGE client60a socket1912 630
119437-119500
START_RANGE client60a socket1913 620
119500-119562
START_RANGE client60a socket1914 630
119562-119625
START_RANGE client60a socket1915 620
119625-119687
START_RANGE client60a socket1916 630
119687-119750
START_RANGE client60a socket1917 620
119750-119812
START_RANGE client60a socket1918 630
119812-119875
START_RANGE client60a socket1919 620
119875-119937
START_RANGE client60a socket1920 630
119937-120000
#elif MASTER_NUM16
START_RANGE client61a socket1921 620
120000-120062
START_RANGE client61a socket1922 630
120062-120125
START_RANGE client61a socket1923 620
120125-120187
START_RANGE client61a socket1924 630
120187-120250
START_RANGE client61a socket1925 620
120250-120312
START_RANGE client61a socket1926 630
120312-120375
START_RANGE client61a socket1927 620
120375-120437
START_RANGE client61a socket1928 630
120437-120500
START_RANGE client61a socket1929 620
120500-120562
START_RANGE client61a socket1930 630
120562-120625
START_RANGE client61a socket1931 620
120625-120687
START_RANGE client61a socket1932 630
120687-120750
START_RANGE client61a socket1933 620
120750-120812
START_RANGE client61a socket1934 630
120812-120875
START_RANGE client61a socket1935 620
120875-120937
START_RANGE client61a socket1936 630
120937-121000
START_RANGE client61a socket1937 620
121000-121062
START_RANGE client61a socket1938 630
121062-121125
START_RANGE client61a socket1939 620
121125-121187
START_RANGE client61a socket1940 630
121187-121250
START_RANGE client61a socket1941 620
121250-121312
START_RANGE client61a socket1942 630
121312-121375
START_RANGE client61a socket1943 620
121375-121437
START_RANGE client61a socket1944 630
121437-121500
START_RANGE client61a socket1945 620
121500-121562
START_RANGE client61a socket1946 630
121562-121625
START_RANGE client61a socket1947 620
121625-121687
START_RANGE client61a socket1948 630
121687-121750
START_RANGE client61a socket1949 620
121750-121812
START_RANGE client61a socket1950 630
121812-121875
START_RANGE client61a socket1951 620
121875-121937
START_RANGE client61a socket1952 630
121937-122000
START_RANGE client62a socket1953 620
122000-122062
START_RANGE client62a socket1954 630
122062-122125
START_RANGE client62a socket1955 620
122125-122187
START_RANGE client62a socket1956 630
122187-122250
START_RANGE client62a socket1957 620
122250-122312
START_RANGE client62a socket1958 630
122312-122375
START_RANGE client62a socket1959 620
122375-122437
START_RANGE client62a socket1960 630
122437-122500
START_RANGE client62a socket1961 620
122500-122562
START_RANGE client62a socket1962 630
122562-122625
```

```
START_RANGE client62a socket1963 620
122625-122687
START_RANGE client62a socket1964 630
122687-122750
START_RANGE client62a socket1965 620
122750-122812
START_RANGE client62a socket1966 630
122812-122875
START_RANGE client62a socket1967 620
122875-122937
START_RANGE client62a socket1968 630
122937-123000
START_RANGE client62a socket1969 620
123000-123062
START_RANGE client62a socket1970 630
123062-123125
START_RANGE client62a socket1971 620
123125-123187
START_RANGE client62a socket1972 630
123187-123250
START_RANGE client62a socket1973 620
123250-123312
START_RANGE client62a socket1974 630
123312-123375
START_RANGE client62a socket1975 620
123375-123437
START_RANGE client62a socket1976 630
123437-123500
START_RANGE client62a socket1977 620
123500-123562
START_RANGE client62a socket1978 630
123562-123625
START_RANGE client62a socket1979 620
123625-123687
START_RANGE client62a socket1980 630
123687-123750
START_RANGE client62a socket1981 620
123750-123812
START_RANGE client62a socket1982 630
123812-123875
START_RANGE client62a socket1983 620
123875-123937
START_RANGE client62a socket1984 630
123937-124000
START_RANGE client63a socket1985 620
124000-124062
START_RANGE client63a socket1986 630
124062-124125
START_RANGE client63a socket1987 620
124125-124187
START_RANGE client63a socket1988 630
124187-124250
START_RANGE client63a socket1989 620
124250-124312
START_RANGE client63a socket1990 630
124312-124375
START_RANGE client63a socket1991 620
124375-124437
START_RANGE client63a socket1992 630
124437-124500
START_RANGE client63a socket1993 620
124500-124562
START_RANGE client63a socket1994 630
124562-124625
START_RANGE client63a socket1995 620
124625-124687
START_RANGE client63a socket1996 630
124687-124750
START_RANGE client63a socket1997 620
124750-124812
START_RANGE client63a socket1998 630
124812-124875
START_RANGE client63a socket1999 620
124875-124937
START_RANGE client63a socket2000 630
124937-125000
START_RANGE client63a socket2001 620
125000-125062
START_RANGE client63a socket2002 630
125062-125125
START_RANGE client63a socket2003 620
125125-125187
START_RANGE client63a socket2004 630
125187-125250
START_RANGE client63a socket2005 620
125250-125312
START_RANGE client63a socket2006 630
125312-125375
START_RANGE client63a socket2007 620
125375-125437
START_RANGE client63a socket2008 630
125437-125500
START_RANGE client63a socket2009 620
125500-125562
START_RANGE client63a socket2010 630
125562-125625
START_RANGE client63a socket2011 620
125625-125687
START_RANGE client63a socket2012 630
125687-125750
START_RANGE client63a socket2013 620
125750-125812
START_RANGE client63a socket2014 630
125812-125875
START_RANGE client63a socket2015 620
125875-125937
START_RANGE client63a socket2016 630
125937-126000
START_RANGE client64a socket2017 620
126000-126062
START_RANGE client64a socket2018 630
126062-126125
START_RANGE client64a socket2019 620
126125-126187
START_RANGE client64a socket2020 630
126187-126250
START_RANGE client64a socket2021 620
126250-126312
START_RANGE client64a socket2022 630
126312-126375
START_RANGE client64a socket2023 620
126375-126437
START_RANGE client64a socket2024 630
126437-126500
START_RANGE client64a socket2025 620
126500-126562
START_RANGE client64a socket2026 630
126562-126625
START_RANGE client64a socket2027 620
126625-126687
START_RANGE client64a socket2028 630
126687-126750
START_RANGE client64a socket2029 620
126750-126812
START_RANGE client64a socket2030 630
126812-126875
START_RANGE client64a socket2031 620
126875-126937
START_RANGE client64a socket2032 630
126937-127000
START_RANGE client64a socket2033 620
127000-127062
START_RANGE client64a socket2034 630
127062-127125
START_RANGE client64a socket2035 620
127125-127187
START_RANGE client64a socket2036 630
127187-127250
START_RANGE client64a socket2037 620
127250-127312
START_RANGE client64a socket2038 630
127312-127375
START_RANGE client64a socket2039 620
127375-127437
START_RANGE client64a socket2040 630
127437-127500
START_RANGE client64a socket2041 620
127500-127562
START_RANGE client64a socket2042 630
127562-127625
START_RANGE client64a socket2043 620
127625-127687
START_RANGE client64a socket2044 630
127687-127750
START_RANGE client64a socket2045 620
127750-127812
START_RANGE client64a socket2046 630
127812-127875
START_RANGE client64a socket2047 620
127875-127937
START_RANGE client64a socket2048 630
127937-128000
#elif MASTER_NUM17
START_RANGE client65a socket2049 620
128000-128062
START_RANGE client65a socket2050 630
128062-128125
START_RANGE client65a socket2051 620
128125-128187
START_RANGE client65a socket2052 630
128187-128250
START_RANGE client65a socket2053 620
128250-128312
START_RANGE client65a socket2054 630
128312-128375
START_RANGE client65a socket2055 620
128375-128437
START_RANGE client65a socket2056 630
128437-128500
START_RANGE client65a socket2057 620
128500-128562
START_RANGE client65a socket2058 630
128562-128625
START_RANGE client65a socket2059 620
128625-128687
START_RANGE client65a socket2060 630
128687-128750
START_RANGE client65a socket2061 620
128750-128812
START_RANGE client65a socket2062 630
128812-128875
START_RANGE client65a socket2063 620
128875-128937
START_RANGE client65a socket2064 630
128937-129000
START_RANGE client65a socket2065 620
129000-129062
START_RANGE client65a socket2066 630
129062-129125
START_RANGE client65a socket2067 620
129125-129187
START_RANGE client65a socket2068 630
129187-129250
START_RANGE client65a socket2069 620
129250-129312
START_RANGE client65a socket2070 630
129312-129375
START_RANGE client65a socket2071 620
129375-129437
START_RANGE client65a socket2072 630
129437-129500
START_RANGE client65a socket2073 620
129500-129562
START_RANGE client65a socket2074 630
129562-129625
START_RANGE client65a socket2075 620
129625-129687
START_RANGE client65a socket2076 630
129687-129750
START_RANGE client65a socket2077 620
129750-129812
START_RANGE client65a socket2078 630
129812-129875
START_RANGE client65a socket2079 620
129875-129937
START_RANGE client65a socket2080 630
129937-130000
START_RANGE client66a socket2081 620
130000-130062
START_RANGE client66a socket2082 630
130062-130125
START_RANGE client66a socket2083 620
130125-130187
START_RANGE client66a socket2084 630
130187-130250
START_RANGE client66a socket2085 620
130250-130312
```

```
START_RANGE client66a socket2086 630
130312-130375
START_RANGE client66a socket2087 620
130375-130437
START_RANGE client66a socket2088 630
130437-130500
START_RANGE client66a socket2089 620
130500-130562
START_RANGE client66a socket2090 630
130562-130625
START_RANGE client66a socket2091 620
130625-130687
START_RANGE client66a socket2092 630
130687-130750
START_RANGE client66a socket2093 620
130750-130812
START_RANGE client66a socket2094 630
130812-130875
START_RANGE client66a socket2095 620
130875-130937
START_RANGE client66a socket2096 630
130937-131000
START_RANGE client66a socket2097 620
131000-131062
START_RANGE client66a socket2098 630
131062-131125
START_RANGE client66a socket2099 620
131125-131187
START_RANGE client66a socket2100 630
131187-131250
START_RANGE client66a socket2101 620
131250-131312
START_RANGE client66a socket2102 630
131312-131375
START_RANGE client66a socket2103 620
131375-131437
START_RANGE client66a socket2104 630
131437-131500
START_RANGE client66a socket2105 620
131500-131562
START_RANGE client66a socket2106 630
131562-131625
START_RANGE client66a socket2107 620
131625-131687
START_RANGE client66a socket2108 630
131687-131750
START_RANGE client66a socket2109 620
131750-131812
START_RANGE client66a socket2110 630
131812-131875
START_RANGE client66a socket2111 620
131875-131937
START_RANGE client66a socket2112 630
131937-132000
START_RANGE client67a socket2113 620
132000-132062
START_RANGE client67a socket2114 630
132062-132125
START_RANGE client67a socket2115 620
132125-132187
START_RANGE client67a socket2116 630
132187-132250
START_RANGE client67a socket2117 620
132250-132312
START_RANGE client67a socket2118 630
132312-132375
START_RANGE client67a socket2119 620
132375-132437
START_RANGE client67a socket2120 630
132437-132500
START_RANGE client67a socket2121 620
132500-132562
START_RANGE client67a socket2122 630
132562-132625
START_RANGE client67a socket2123 620
132625-132687
START_RANGE client67a socket2124 630
132687-132750
START_RANGE client67a socket2125 620
132750-132812
START_RANGE client67a socket2126 630
132812-132875
START_RANGE client67a socket2127 620
132875-132937
START_RANGE client67a socket2128 630
132937-133000
START_RANGE client67a socket2129 620
133000-133062
START_RANGE client67a socket2130 630
133062-133125
START_RANGE client67a socket2131 620
133125-133187
START_RANGE client67a socket2132 630
133187-133250
START_RANGE client67a socket2133 620
133250-133312
START_RANGE client67a socket2134 630
133312-133375
START_RANGE client67a socket2135 620
133375-133437
START_RANGE client67a socket2136 630
133437-133500
START_RANGE client67a socket2137 620
133500-133562
START_RANGE client67a socket2138 630
133562-133625
START_RANGE client67a socket2139 620
133625-133687
START_RANGE client67a socket2140 630
133687-133750
START_RANGE client67a socket2141 620
133750-133812
START_RANGE client67a socket2142 630
133812-133875
START_RANGE client67a socket2143 620
133875-133937
START_RANGE client67a socket2144 630
133937-134000
START_RANGE client68a socket2145 620
134000-134062
START_RANGE client68a socket2146 630
134062-134125
START_RANGE client68a socket2147 620
134125-134187
START_RANGE client68a socket2148 630
134187-134250
START_RANGE client68a socket2149 620
134250-134312
START_RANGE client68a socket2150 630
134312-134375
START_RANGE client68a socket2151 620
134375-134437
START_RANGE client68a socket2152 630
134437-134500
START_RANGE client68a socket2153 620
134500-134562
START_RANGE client68a socket2154 630
134562-134625
START_RANGE client68a socket2155 620
134625-134687
START_RANGE client68a socket2156 630
134687-134750
START_RANGE client68a socket2157 620
134750-134812
START_RANGE client68a socket2158 630
134812-134875
START_RANGE client68a socket2159 620
134875-134937
START_RANGE client68a socket2160 630
134937-135000
START_RANGE client68a socket2161 620
135000-135062
START_RANGE client68a socket2162 630
135062-135125
START_RANGE client68a socket2163 620
135125-135187
START_RANGE client68a socket2164 630
135187-135250
START_RANGE client68a socket2165 620
135250-135312
START_RANGE client68a socket2166 630
135312-135375
START_RANGE client68a socket2167 620
135375-135437
START_RANGE client68a socket2168 630
135437-135500
START_RANGE client68a socket2169 620
135500-135562
START_RANGE client68a socket2170 630
135562-135625
START_RANGE client68a socket2171 620
135625-135687
START_RANGE client68a socket2172 630
135687-135750
START_RANGE client68a socket2173 620
135750-135812
START_RANGE client68a socket2174 630
135812-135875
START_RANGE client68a socket2175 620
135875-135937
START_RANGE client68a socket2176 630
135937-136000
#elif MASTER_NUM18
START_RANGE client69a socket2177 620
136000-136062
START_RANGE client69a socket2178 630
136062-136125
START_RANGE client69a socket2179 620
136125-136187
START_RANGE client69a socket2180 630
136187-136250
START_RANGE client69a socket2181 620
136250-136312
START_RANGE client69a socket2182 630
136312-136375
START_RANGE client69a socket2183 620
136375-136437
START_RANGE client69a socket2184 630
136437-136500
START_RANGE client69a socket2185 620
136500-136562
START_RANGE client69a socket2186 630
136562-136625
START_RANGE client69a socket2187 620
136625-136687
START_RANGE client69a socket2188 630
136687-136750
START_RANGE client69a socket2189 620
136750-136812
START_RANGE client69a socket2190 630
136812-136875
START_RANGE client69a socket2191 620
136875-136937
START_RANGE client69a socket2192 630
136937-137000
START_RANGE client69a socket2193 620
137000-137062
START_RANGE client69a socket2194 630
137062-137125
START_RANGE client69a socket2195 620
137125-137187
START_RANGE client69a socket2196 630
137187-137250
START_RANGE client69a socket2197 620
137250-137312
START_RANGE client69a socket2198 630
137312-137375
START_RANGE client69a socket2199 620
137375-137437
START_RANGE client69a socket2200 630
137437-137500
START_RANGE client69a socket2201 620
137500-137562
START_RANGE client69a socket2202 630
137562-137625
START_RANGE client69a socket2203 620
137625-137687
START_RANGE client69a socket2204 630
137687-137750
START_RANGE client69a socket2205 620
137750-137812
START_RANGE client69a socket2206 630
137812-137875
START_RANGE client69a socket2207 620
137875-137937
START_RANGE client69a socket2208 630
137937-138000
```

START_RANGE client70a socket2209 620
138000-138062
START_RANGE client70a socket2210 630
138062-138125
START_RANGE client70a socket2211 620
138125-138187
START_RANGE client70a socket2212 630
138187-138250
START_RANGE client70a socket2213 620
138250-138312
START_RANGE client70a socket2214 630
138312-138375
START_RANGE client70a socket2215 620
138375-138437
START_RANGE client70a socket2216 630
138437-138500
START_RANGE client70a socket2217 620
138500-138562
START_RANGE client70a socket2218 630
138562-138625
START_RANGE client70a socket2219 620
138625-138687
START_RANGE client70a socket2220 630
138687-138750
START_RANGE client70a socket2221 620
138750-138812
START_RANGE client70a socket2222 630
138812-138875
START_RANGE client70a socket2223 620
138875-138937
START_RANGE client70a socket2224 630
138937-139000
START_RANGE client70a socket2225 620
139000-139062
START_RANGE client70a socket2226 630
139062-139125
START_RANGE client70a socket2227 620
139125-139187
START_RANGE client70a socket2228 630
139187-139250
START_RANGE client70a socket2229 620
139250-139312
START_RANGE client70a socket2230 630
139312-139375
START_RANGE client70a socket2231 620
139375-139437
START_RANGE client70a socket2232 630
139437-139500
START_RANGE client70a socket2233 620
139500-139562
START_RANGE client70a socket2234 630
139562-139625
START_RANGE client70a socket2235 620
139625-139687
START_RANGE client70a socket2236 630
139687-139750
START_RANGE client70a socket2237 620
139750-139812
START_RANGE client70a socket2238 630
139812-139875
START_RANGE client70a socket2239 620
139875-139937
START_RANGE client70a socket2240 630
139937-140000
START_RANGE client71a socket2241 620
140000-140062
START_RANGE client71a socket2242 630
140062-140125
START_RANGE client71a socket2243 620
140125-140187
START_RANGE client71a socket2244 630
140187-140250
START_RANGE client71a socket2245 620
140250-140312
START_RANGE client71a socket2246 630
140312-140375
START_RANGE client71a socket2247 620
140375-140437
START_RANGE client71a socket2248 630
140437-140500
START_RANGE client71a socket2249 620
140500-140562
START_RANGE client71a socket2250 630
140562-140625
START_RANGE client71a socket2251 620
140625-140687
START_RANGE client71a socket2252 630
140687-140750
START_RANGE client71a socket2253 620
140750-140812
START_RANGE client71a socket2254 630
140812-140875
START_RANGE client71a socket2255 620
140875-140937
START_RANGE client71a socket2256 630
140937-141000
START_RANGE client71a socket2257 620
141000-141062
START_RANGE client71a socket2258 630
141062-141125
START_RANGE client71a socket2259 620
141125-141187
START_RANGE client71a socket2260 630
141187-141250
START_RANGE client71a socket2261 620
141250-141312
START_RANGE client71a socket2262 630
141312-141375
START_RANGE client71a socket2263 620
141375-141437
START_RANGE client71a socket2264 630
141437-141500
START_RANGE client71a socket2265 620
141500-141562
START_RANGE client71a socket2266 630
141562-141625
START_RANGE client71a socket2267 620
141625-141687
START_RANGE client71a socket2268 630
141687-141750
START_RANGE client71a socket2269 620
141750-141812
START_RANGE client71a socket2270 630
141812-141875
START_RANGE client71a socket2271 620
141875-141937
START_RANGE client71a socket2272 630
141937-142000
START_RANGE client72a socket2273 620
142000-142062
START_RANGE client72a socket2274 630
142062-142125
START_RANGE client72a socket2275 620
142125-142187
START_RANGE client72a socket2276 630
142187-142250
START_RANGE client72a socket2277 620
142250-142312
START_RANGE client72a socket2278 630
142312-142375
START_RANGE client72a socket2279 620
142375-142437
START_RANGE client72a socket2280 630
142437-142500
START_RANGE client72a socket2281 620
142500-142562
START_RANGE client72a socket2282 630
142562-142625
START_RANGE client72a socket2283 620
142625-142687
START_RANGE client72a socket2284 630
142687-142750
START_RANGE client72a socket2285 620
142750-142812
START_RANGE client72a socket2286 630
142812-142875
START_RANGE client72a socket2287 620
142875-142937
START_RANGE client72a socket2288 630
142937-143000
START_RANGE client72a socket2289 620
143000-143062
START_RANGE client72a socket2290 630
143062-143125
START_RANGE client72a socket2291 620
143125-143187
START_RANGE client72a socket2292 630
143187-143250
START_RANGE client72a socket2293 620
143250-143312
START_RANGE client72a socket2294 630
143312-143375
START_RANGE client72a socket2295 620
143375-143437
START_RANGE client72a socket2296 630
143437-143500
START_RANGE client72a socket2297 620
143500-143562
START_RANGE client72a socket2298 630
143562-143625
START_RANGE client72a socket2299 620
143625-143687
START_RANGE client72a socket2300 630
143687-143750
START_RANGE client72a socket2301 620
143750-143812
START_RANGE client72a socket2302 630
143812-143875
START_RANGE client72a socket2303 620
143875-143937
START_RANGE client72a socket2304 630
143937-144000
#elif MASTER_NUM19
START_RANGE client73a socket2305 620
144000-144062
START_RANGE client73a socket2306 630
144062-144125
START_RANGE client73a socket2307 620
144125-144187
START_RANGE client73a socket2308 630
144187-144250
START_RANGE client73a socket2309 620
144250-144312
START_RANGE client73a socket2310 630
144312-144375
START_RANGE client73a socket2311 620
144375-144437
START_RANGE client73a socket2312 630
144437-144500
START_RANGE client73a socket2313 620
144500-144562
START_RANGE client73a socket2314 630
144562-144625
START_RANGE client73a socket2315 620
144625-144687
START_RANGE client73a socket2316 630
144687-144750
START_RANGE client73a socket2317 620
144750-144812
START_RANGE client73a socket2318 630
144812-144875
START_RANGE client73a socket2319 620
144875-144937
START_RANGE client73a socket2320 630
144937-145000
START_RANGE client73a socket2321 620
145000-145062
START_RANGE client73a socket2322 630
145062-145125
START_RANGE client73a socket2323 620
145125-145187
START_RANGE client73a socket2324 630
145187-145250
START_RANGE client73a socket2325 620
145250-145312
START_RANGE client73a socket2326 630
145312-145375
START_RANGE client73a socket2327 620
145375-145437
START_RANGE client73a socket2328 630
145437-145500
START_RANGE client73a socket2329 620
145500-145562
START_RANGE client73a socket2330 630
145562-145625
START_RANGE client73a socket2331 620
145625-145687

START_RANGE client73a socket2332 630
145687-145750
START_RANGE client73a socket2333 620
145750-145812
START_RANGE client73a socket2334 630
145812-145875
START_RANGE client73a socket2335 620
145875-145937
START_RANGE client73a socket2336 630
145937-146000
START_RANGE client74a socket2337 620
146000-146062
START_RANGE client74a socket2338 630
146062-146125
START_RANGE client74a socket2339 620
146125-146187
START_RANGE client74a socket2340 630
146187-146250
START_RANGE client74a socket2341 620
146250-146312
START_RANGE client74a socket2342 630
146312-146375
START_RANGE client74a socket2343 620
146375-146437
START_RANGE client74a socket2344 630
146437-146500
START_RANGE client74a socket2345 620
146500-146562
START_RANGE client74a socket2346 630
146562-146625
START_RANGE client74a socket2347 620
146625-146687
START_RANGE client74a socket2348 630
146687-146750
START_RANGE client74a socket2349 620
146750-146812
START_RANGE client74a socket2350 630
146812-146875
START_RANGE client74a socket2351 620
146875-146937
START_RANGE client74a socket2352 630
146937-147000
START_RANGE client74a socket2353 620
147000-147062
START_RANGE client74a socket2354 630
147062-147125
START_RANGE client74a socket2355 620
147125-147187
START_RANGE client74a socket2356 630
147187-147250
START_RANGE client74a socket2357 620
147250-147312
START_RANGE client74a socket2358 630
147312-147375
START_RANGE client74a socket2359 620
147375-147437
START_RANGE client74a socket2360 630
147437-147500
START_RANGE client74a socket2361 620
147500-147562
START_RANGE client74a socket2362 630
147562-147625
START_RANGE client74a socket2363 620
147625-147687
START_RANGE client74a socket2364 630
147687-147750
START_RANGE client74a socket2365 620
147750-147812
START_RANGE client74a socket2366 630
147812-147875
START_RANGE client74a socket2367 620
147875-147937
START_RANGE client74a socket2368 630
147937-148000
START_RANGE client75a socket2369 620
148000-148062
START_RANGE client75a socket2370 630
148062-148125
START_RANGE client75a socket2371 620
148125-148187
START_RANGE client75a socket2372 630
148187-148250
START_RANGE client75a socket2373 620
148250-148312
START_RANGE client75a socket2374 630
148312-148375
START_RANGE client75a socket2375 620
148375-148437
START_RANGE client75a socket2376 630
148437-148500
START_RANGE client75a socket2377 620
148500-148562
START_RANGE client75a socket2378 630
148562-148625
START_RANGE client75a socket2379 620
148625-148687
START_RANGE client75a socket2380 630
148687-148750
START_RANGE client75a socket2381 620
148750-148812
START_RANGE client75a socket2382 630
148812-148875
START_RANGE client75a socket2383 620
148875-148937
START_RANGE client75a socket2384 630
148937-149000
START_RANGE client75a socket2385 620
149000-149062
START_RANGE client75a socket2386 630
149062-149125
START_RANGE client75a socket2387 620
149125-149187
START_RANGE client75a socket2388 630
149187-149250
START_RANGE client75a socket2389 620
149250-149312
START_RANGE client75a socket2390 630
149312-149375
START_RANGE client75a socket2391 620
149375-149437
START_RANGE client75a socket2392 630
149437-149500
START_RANGE client75a socket2393 620
149500-149562
START_RANGE client75a socket2394 630
149562-149625
START_RANGE client75a socket2395 620
149625-149687
START_RANGE client75a socket2396 630
149687-149750
START_RANGE client75a socket2397 620
149750-149812
START_RANGE client75a socket2398 630
149812-149875
START_RANGE client75a socket2399 620
149875-149937
START_RANGE client75a socket2400 630
149937-150000
START_RANGE client76a socket2401 620
150000-150062
START_RANGE client76a socket2402 630
150062-150125
START_RANGE client76a socket2403 620
150125-150187
START_RANGE client76a socket2404 630
150187-150250
START_RANGE client76a socket2405 620
150250-150312
START_RANGE client76a socket2406 630
150312-150375
START_RANGE client76a socket2407 620
150375-150437
START_RANGE client76a socket2408 630
150437-150500
START_RANGE client76a socket2409 620
150500-150562
START_RANGE client76a socket2410 630
150562-150625
START_RANGE client76a socket2411 620
150625-150687
START_RANGE client76a socket2412 630
150687-150750
START_RANGE client76a socket2413 620
150750-150812
START_RANGE client76a socket2414 630
150812-150875
START_RANGE client76a socket2415 620
150875-150937
START_RANGE client76a socket2416 630
150937-151000
START_RANGE client76a socket2417 620
151000-151062
START_RANGE client76a socket2418 630
151062-151125
START_RANGE client76a socket2419 620
151125-151187
START_RANGE client76a socket2420 630
151187-151250
START_RANGE client76a socket2421 620
151250-151312
START_RANGE client76a socket2422 630
151312-151375
START_RANGE client76a socket2423 620
151375-151437
START_RANGE client76a socket2424 630
151437-151500
START_RANGE client76a socket2425 620
151500-151562
START_RANGE client76a socket2426 630
151562-151625
START_RANGE client76a socket2427 620
151625-151687
START_RANGE client76a socket2428 630
151687-151750
START_RANGE client76a socket2429 620
151750-151812
START_RANGE client76a socket2430 630
151812-151875
START_RANGE client76a socket2431 620
151875-151937
START_RANGE client76a socket2432 630
151937-152000
#elif MASTER_NUM20
START_RANGE client77a socket2433 620
152000-152062
START_RANGE client77a socket2434 630
152062-152125
START_RANGE client77a socket2435 620
152125-152187
START_RANGE client77a socket2436 630
152187-152250
START_RANGE client77a socket2437 620
152250-152312
START_RANGE client77a socket2438 630
152312-152375
START_RANGE client77a socket2439 620
152375-152437
START_RANGE client77a socket2440 630
152437-152500
START_RANGE client77a socket2441 620
152500-152562
START_RANGE client77a socket2442 630
152562-152625
START_RANGE client77a socket2443 620
152625-152687
START_RANGE client77a socket2444 630
152687-152750
START_RANGE client77a socket2445 620
152750-152812
START_RANGE client77a socket2446 630
152812-152875
START_RANGE client77a socket2447 620
152875-152937
START_RANGE client77a socket2448 630
152937-153000
START_RANGE client77a socket2449 620
153000-153062
START_RANGE client77a socket2450 630
153062-153125
START_RANGE client77a socket2451 620
153125-153187
START_RANGE client77a socket2452 630
153187-153250
START_RANGE client77a socket2453 620
153250-153312
START_RANGE client77a socket2454 630
153312-153375

```
START_RANGE client77a socket2455 620
153375-153437
START_RANGE client77a socket2456 630
153437-153500
START_RANGE client77a socket2457 620
153500-153562
START_RANGE client77a socket2458 630
153562-153625
START_RANGE client77a socket2459 620
153625-153687
START_RANGE client77a socket2460 630
153687-153750
START_RANGE client77a socket2461 620
153750-153812
START_RANGE client77a socket2462 630
153812-153875
START_RANGE client77a socket2463 620
153875-153937
START_RANGE client77a socket2464 630
153937-154000
START_RANGE client78a socket2465 620
154000-154062
START_RANGE client78a socket2466 630
154062-154125
START_RANGE client78a socket2467 620
154125-154187
START_RANGE client78a socket2468 630
154187-154250
START_RANGE client78a socket2469 620
154250-154312
START_RANGE client78a socket2470 630
154312-154375
START_RANGE client78a socket2471 620
154375-154437
START_RANGE client78a socket2472 630
154437-154500
START_RANGE client78a socket2473 620
154500-154562
START_RANGE client78a socket2474 630
154562-154625
START_RANGE client78a socket2475 620
154625-154687
START_RANGE client78a socket2476 630
154687-154750
START_RANGE client78a socket2477 620
154750-154812
START_RANGE client78a socket2478 630
154812-154875
START_RANGE client78a socket2479 620
154875-154937
START_RANGE client78a socket2480 630
154937-155000
START_RANGE client78a socket2481 620
155000-155062
START_RANGE client78a socket2482 630
155062-155125
START_RANGE client78a socket2483 620
155125-155187
START_RANGE client78a socket2484 630
155187-155250
START_RANGE client78a socket2485 620
155250-155312
START_RANGE client78a socket2486 630
155312-155375
START_RANGE client78a socket2487 620
155375-155437
START_RANGE client78a socket2488 630
155437-155500
START_RANGE client78a socket2489 620
155500-155562
START_RANGE client78a socket2490 630
155562-155625
START_RANGE client78a socket2491 620
155625-155687
START_RANGE client78a socket2492 630
155687-155750
START_RANGE client78a socket2493 620
155750-155812
START_RANGE client78a socket2494 630
155812-155875
START_RANGE client78a socket2495 620
155875-155937
START_RANGE client78a socket2496 630
155937-156000
START_RANGE client79a socket2497 620
156000-156062
START_RANGE client79a socket2498 630
156062-156125
START_RANGE client79a socket2499 620
156125-156187
START_RANGE client79a socket2500 630
156187-156250
START_RANGE client79a socket2501 620
156250-156312
START_RANGE client79a socket2502 630
156312-156375
START_RANGE client79a socket2503 620
156375-156437
START_RANGE client79a socket2504 630
156437-156500
START_RANGE client79a socket2505 620
156500-156562
START_RANGE client79a socket2506 630
156562-156625
START_RANGE client79a socket2507 620
156625-156687
START_RANGE client79a socket2508 630
156687-156750
START_RANGE client79a socket2509 620
156750-156812
START_RANGE client79a socket2510 630
156812-156875
START_RANGE client79a socket2511 620
156875-156937
START_RANGE client79a socket2512 630
156937-157000
START_RANGE client79a socket2513 620
157000-157062
START_RANGE client79a socket2514 630
157062-157125
START_RANGE client79a socket2515 620
157125-157187
START_RANGE client79a socket2516 630
157187-157250
START_RANGE client79a socket2517 620
157250-157312
START_RANGE client79a socket2518 630
157312-157375
START_RANGE client79a socket2519 620
157375-157437
START_RANGE client79a socket2520 630
157437-157500
START_RANGE client79a socket2521 620
157500-157562
START_RANGE client79a socket2522 630
157562-157625
START_RANGE client79a socket2523 620
157625-157687
START_RANGE client79a socket2524 630
157687-157750
START_RANGE client79a socket2525 620
157750-157812
START_RANGE client79a socket2526 630
157812-157875
START_RANGE client79a socket2527 620
157875-157937
START_RANGE client79a socket2528 630
157937-158000
START_RANGE client80a socket2529 620
158000-158062
START_RANGE client80a socket2530 630
158062-158125
START_RANGE client80a socket2531 620
158125-158187
START_RANGE client80a socket2532 630
158187-158250
START_RANGE client80a socket2533 620
158250-158312
START_RANGE client80a socket2534 630
158312-158375
START_RANGE client80a socket2535 620
158375-158437
START_RANGE client80a socket2536 630
158437-158500
START_RANGE client80a socket2537 620
158500-158562
START_RANGE client80a socket2538 630
158562-158625
START_RANGE client80a socket2539 620
158625-158687
START_RANGE client80a socket2540 630
158687-158750
START_RANGE client80a socket2541 620
158750-158812
START_RANGE client80a socket2542 630
158812-158875
START_RANGE client80a socket2543 620
158875-158937
START_RANGE client80a socket2544 630
158937-159000
START_RANGE client80a socket2545 620
159000-159062
START_RANGE client80a socket2546 630
159062-159125
START_RANGE client80a socket2547 620
159125-159187
START_RANGE client80a socket2548 630
159187-159250
START_RANGE client80a socket2549 620
159250-159312
START_RANGE client80a socket2550 630
159312-159375
START_RANGE client80a socket2551 620
159375-159437
START_RANGE client80a socket2552 630
159437-159500
START_RANGE client80a socket2553 620
159500-159562
START_RANGE client80a socket2554 630
159562-159625
START_RANGE client80a socket2555 620
159625-159687
START_RANGE client80a socket2556 630
159687-159750
START_RANGE client80a socket2557 620
159750-159812
START_RANGE client80a socket2558 630
159812-159875
START_RANGE client80a socket2559 620
159875-159937
START_RANGE client80a socket2560 630
159937-160000
#elif MASTER_NUM21
START_RANGE client81a socket2561 620
160000-160062
START_RANGE client81a socket2562 630
160062-160125
START_RANGE client81a socket2563 620
160125-160187
START_RANGE client81a socket2564 630
160187-160250
START_RANGE client81a socket2565 620
160250-160312
START_RANGE client81a socket2566 630
160312-160375
START_RANGE client81a socket2567 620
160375-160437
START_RANGE client81a socket2568 630
160437-160500
START_RANGE client81a socket2569 620
160500-160562
START_RANGE client81a socket2570 630
160562-160625
START_RANGE client81a socket2571 620
160625-160687
START_RANGE client81a socket2572 630
160687-160750
START_RANGE client81a socket2573 620
160750-160812
START_RANGE client81a socket2574 630
160812-160875
START_RANGE client81a socket2575 620
160875-160937
START_RANGE client81a socket2576 630
160937-161000
START_RANGE client81a socket2577 620
161000-161062
```

```
START_RANGE client81a socket2578 630
161062-161125
START_RANGE client81a socket2579 620
161125-161187
START_RANGE client81a socket2580 630
161187-161250
START_RANGE client81a socket2581 620
161250-161312
START_RANGE client81a socket2582 630
161312-161375
START_RANGE client81a socket2583 620
161375-161437
START_RANGE client81a socket2584 630
161437-161500
START_RANGE client81a socket2585 620
161500-161562
START_RANGE client81a socket2586 630
161562-161625
START_RANGE client81a socket2587 620
161625-161687
START_RANGE client81a socket2588 630
161687-161750
START_RANGE client81a socket2589 620
161750-161812
START_RANGE client81a socket2590 630
161812-161875
START_RANGE client81a socket2591 620
161875-161937
START_RANGE client81a socket2592 630
161937-162000
START_RANGE client82a socket2593 620
162000-162062
START_RANGE client82a socket2594 630
162062-162125
START_RANGE client82a socket2595 620
162125-162187
START_RANGE client82a socket2596 630
162187-162250
START_RANGE client82a socket2597 620
162250-162312
START_RANGE client82a socket2598 630
162312-162375
START_RANGE client82a socket2599 620
162375-162437
START_RANGE client82a socket2600 630
162437-162500
START_RANGE client82a socket2601 620
162500-162562
START_RANGE client82a socket2602 630
162562-162625
START_RANGE client82a socket2603 620
162625-162687
START_RANGE client82a socket2604 630
162687-162750
START_RANGE client82a socket2605 620
162750-162812
START_RANGE client82a socket2606 630
162812-162875
START_RANGE client82a socket2607 620
162875-162937
START_RANGE client82a socket2608 630
162937-163000
START_RANGE client82a socket2609 620
163000-163062
START_RANGE client82a socket2610 630
163062-163125
START_RANGE client82a socket2611 620
163125-163187
START_RANGE client82a socket2612 630
163187-163250
START_RANGE client82a socket2613 620
163250-163312
START_RANGE client82a socket2614 630
163312-163375
START_RANGE client82a socket2615 620
163375-163437
START_RANGE client82a socket2616 630
163437-163500
START_RANGE client82a socket2617 620
163500-163562
START_RANGE client82a socket2618 630
163562-163625
START_RANGE client82a socket2619 620
163625-163687
START_RANGE client82a socket2620 630
163687-163750
START_RANGE client82a socket2621 620
163750-163812
START_RANGE client82a socket2622 630
163812-163875
START_RANGE client82a socket2623 620
163875-163937
START_RANGE client82a socket2624 630
163937-164000
START_RANGE client83a socket2625 620
164000-164062
START_RANGE client83a socket2626 630
164062-164125
START_RANGE client83a socket2627 620
164125-164187
START_RANGE client83a socket2628 630
164187-164250
START_RANGE client83a socket2629 620
164250-164312
START_RANGE client83a socket2630 630
164312-164375
START_RANGE client83a socket2631 620
164375-164437
START_RANGE client83a socket2632 630
164437-164500
START_RANGE client83a socket2633 620
164500-164562
START_RANGE client83a socket2634 630
164562-164625
START_RANGE client83a socket2635 620
164625-164687
START_RANGE client83a socket2636 630
164687-164750
START_RANGE client83a socket2637 620
164750-164812
START_RANGE client83a socket2638 630
164812-164875
START_RANGE client83a socket2639 620
164875-164937
START_RANGE client83a socket2640 630
164937-165000
START_RANGE client83a socket2641 620
165000-165062
START_RANGE client83a socket2642 630
165062-165125
START_RANGE client83a socket2643 620
165125-165187
START_RANGE client83a socket2644 630
165187-165250
START_RANGE client83a socket2645 620
165250-165312
START_RANGE client83a socket2646 630
165312-165375
START_RANGE client83a socket2647 620
165375-165437
START_RANGE client83a socket2648 630
165437-165500
START_RANGE client83a socket2649 620
165500-165562
START_RANGE client83a socket2650 630
165562-165625
START_RANGE client83a socket2651 620
165625-165687
START_RANGE client83a socket2652 630
165687-165750
START_RANGE client83a socket2653 620
165750-165812
START_RANGE client83a socket2654 630
165812-165875
START_RANGE client83a socket2655 620
165875-165937
START_RANGE client83a socket2656 630
165937-166000
START_RANGE client84a socket2657 620
166000-166062
START_RANGE client84a socket2658 630
166062-166125
START_RANGE client84a socket2659 620
166125-166187
START_RANGE client84a socket2660 630
166187-166250
START_RANGE client84a socket2661 620
166250-166312
START_RANGE client84a socket2662 630
166312-166375
START_RANGE client84a socket2663 620
166375-166437
START_RANGE client84a socket2664 630
166437-166500
START_RANGE client84a socket2665 620
166500-166562
START_RANGE client84a socket2666 630
166562-166625
START_RANGE client84a socket2667 620
166625-166687
START_RANGE client84a socket2668 630
166687-166750
START_RANGE client84a socket2669 620
166750-166812
START_RANGE client84a socket2670 630
166812-166875
START_RANGE client84a socket2671 620
166875-166937
START_RANGE client84a socket2672 630
166937-167000
START_RANGE client84a socket2673 620
167000-167062
START_RANGE client84a socket2674 630
167062-167125
START_RANGE client84a socket2675 620
167125-167187
START_RANGE client84a socket2676 630
167187-167250
START_RANGE client84a socket2677 620
167250-167312
START_RANGE client84a socket2678 630
167312-167375
START_RANGE client84a socket2679 620
167375-167437
START_RANGE client84a socket2680 630
167437-167500
START_RANGE client84a socket2681 620
167500-167562
START_RANGE client84a socket2682 630
167562-167625
START_RANGE client84a socket2683 620
167625-167687
START_RANGE client84a socket2684 630
167687-167750
START_RANGE client84a socket2685 620
167750-167812
START_RANGE client84a socket2686 630
167812-167875
START_RANGE client84a socket2687 620
167875-167937
START_RANGE client84a socket2688 630
167937-168000
#elif MASTER_NUM22
START_RANGE client85a socket2689 620
168000-168062
START_RANGE client85a socket2690 630
168062-168125
START_RANGE client85a socket2691 620
168125-168187
START_RANGE client85a socket2692 630
168187-168250
START_RANGE client85a socket2693 620
168250-168312
START_RANGE client85a socket2694 630
168312-168375
START_RANGE client85a socket2695 620
168375-168437
START_RANGE client85a socket2696 630
168437-168500
START_RANGE client85a socket2697 620
168500-168562
START_RANGE client85a socket2698 630
168562-168625
START_RANGE client85a socket2699 620
168625-168687
START_RANGE client85a socket2700 630
168687-168750
```

```
START_RANGE client85a socket2701 620
168750-168812
START_RANGE client85a socket2702 630
168812-168875
START_RANGE client85a socket2703 620
168875-168937
START_RANGE client85a socket2704 630
168937-169000
START_RANGE client85a socket2705 620
169000-169062
START_RANGE client85a socket2706 630
169062-169125
START_RANGE client85a socket2707 620
169125-169187
START_RANGE client85a socket2708 630
169187-169250
START_RANGE client85a socket2709 620
169250-169312
START_RANGE client85a socket2710 630
169312-169375
START_RANGE client85a socket2711 620
169375-169437
START_RANGE client85a socket2712 630
169437-169500
START_RANGE client85a socket2713 620
169500-169562
START_RANGE client85a socket2714 630
169562-169625
START_RANGE client85a socket2715 620
169625-169687
START_RANGE client85a socket2716 630
169687-169750
START_RANGE client85a socket2717 620
169750-169812
START_RANGE client85a socket2718 630
169812-169875
START_RANGE client85a socket2719 620
169875-169937
START_RANGE client85a socket2720 630
169937-170000
START_RANGE client86a socket2721 620
170000-170062
START_RANGE client86a socket2722 630
170062-170125
START_RANGE client86a socket2723 620
170125-170187
START_RANGE client86a socket2724 630
170187-170250
START_RANGE client86a socket2725 620
170250-170312
START_RANGE client86a socket2726 630
170312-170375
START_RANGE client86a socket2727 620
170375-170437
START_RANGE client86a socket2728 630
170437-170500
START_RANGE client86a socket2729 620
170500-170562
START_RANGE client86a socket2730 630
170562-170625
START_RANGE client86a socket2731 620
170625-170687
START_RANGE client86a socket2732 630
170687-170750
START_RANGE client86a socket2733 620
170750-170812
START_RANGE client86a socket2734 630
170812-170875
START_RANGE client86a socket2735 620
170875-170937
START_RANGE client86a socket2736 630
170937-171000
START_RANGE client86a socket2737 620
171000-171062
START_RANGE client86a socket2738 630
171062-171125
START_RANGE client86a socket2739 620
171125-171187
START_RANGE client86a socket2740 630
171187-171250
START_RANGE client86a socket2741 620
171250-171312
START_RANGE client86a socket2742 630
171312-171375
START_RANGE client86a socket2743 620
171375-171437
START_RANGE client86a socket2744 630
171437-171500
START_RANGE client86a socket2745 620
171500-171562
START_RANGE client86a socket2746 630
171562-171625
START_RANGE client86a socket2747 620
171625-171687
START_RANGE client86a socket2748 630
171687-171750
START_RANGE client86a socket2749 620
171750-171812
START_RANGE client86a socket2750 630
171812-171875
START_RANGE client86a socket2751 620
171875-171937
START_RANGE client86a socket2752 630
171937-172000
START_RANGE client87a socket2753 620
172000-172062
START_RANGE client87a socket2754 630
172062-172125
START_RANGE client87a socket2755 620
172125-172187
START_RANGE client87a socket2756 630
172187-172250
START_RANGE client87a socket2757 620
172250-172312
START_RANGE client87a socket2758 630
172312-172375
START_RANGE client87a socket2759 620
172375-172437
START_RANGE client87a socket2760 630
172437-172500
START_RANGE client87a socket2761 620
172500-172562
START_RANGE client87a socket2762 630
172562-172625
START_RANGE client87a socket2763 620
172625-172687
START_RANGE client87a socket2764 630
172687-172750
START_RANGE client87a socket2765 620
172750-172812
START_RANGE client87a socket2766 630
172812-172875
START_RANGE client87a socket2767 620
172875-172937
START_RANGE client87a socket2768 630
172937-173000
START_RANGE client87a socket2769 620
173000-173062
START_RANGE client87a socket2770 630
173062-173125
START_RANGE client87a socket2771 620
173125-173187
START_RANGE client87a socket2772 630
173187-173250
START_RANGE client87a socket2773 620
173250-173312
START_RANGE client87a socket2774 630
173312-173375
START_RANGE client87a socket2775 620
173375-173437
START_RANGE client87a socket2776 630
173437-173500
START_RANGE client87a socket2777 620
173500-173562
START_RANGE client87a socket2778 630
173562-173625
START_RANGE client87a socket2779 620
173625-173687
START_RANGE client87a socket2780 630
173687-173750
START_RANGE client87a socket2781 620
173750-173812
START_RANGE client87a socket2782 630
173812-173875
START_RANGE client87a socket2783 620
173875-173937
START_RANGE client87a socket2784 630
173937-174000
START_RANGE client88a socket2785 620
174000-174062
START_RANGE client88a socket2786 630
174062-174125
START_RANGE client88a socket2787 620
174125-174187
START_RANGE client88a socket2788 630
174187-174250
START_RANGE client88a socket2789 620
174250-174312
START_RANGE client88a socket2790 630
174312-174375
START_RANGE client88a socket2791 620
174375-174437
START_RANGE client88a socket2792 630
174437-174500
START_RANGE client88a socket2793 620
174500-174562
START_RANGE client88a socket2794 630
174562-174625
START_RANGE client88a socket2795 620
174625-174687
START_RANGE client88a socket2796 630
174687-174750
START_RANGE client88a socket2797 620
174750-174812
START_RANGE client88a socket2798 630
174812-174875
START_RANGE client88a socket2799 620
174875-174937
START_RANGE client88a socket2800 630
174937-175000
START_RANGE client88a socket2801 620
175000-175062
START_RANGE client88a socket2802 630
175062-175125
START_RANGE client88a socket2803 620
175125-175187
START_RANGE client88a socket2804 630
175187-175250
START_RANGE client88a socket2805 620
175250-175312
START_RANGE client88a socket2806 630
175312-175375
START_RANGE client88a socket2807 620
175375-175437
START_RANGE client88a socket2808 630
175437-175500
START_RANGE client88a socket2809 620
175500-175562
START_RANGE client88a socket2810 630
175562-175625
START_RANGE client88a socket2811 620
175625-175687
START_RANGE client88a socket2812 630
175687-175750
START_RANGE client88a socket2813 620
175750-175812
START_RANGE client88a socket2814 630
175812-175875
START_RANGE client88a socket2815 620
175875-175937
START_RANGE client88a socket2816 630
175937-176000
#elif MASTER_NUM23
START_RANGE client89a socket2817 620
176000-176062
START_RANGE client89a socket2818 630
176062-176125
START_RANGE client89a socket2819 620
176125-176187
START_RANGE client89a socket2820 630
176187-176250
START_RANGE client89a socket2821 620
176250-176312
START_RANGE client89a socket2822 630
176312-176375
START_RANGE client89a socket2823 620
176375-176437
```

```
START_RANGE client89a socket2824 630
176437-176500
START_RANGE client89a socket2825 620
176500-176562
START_RANGE client89a socket2826 630
176562-176625
START_RANGE client89a socket2827 620
176625-176687
START_RANGE client89a socket2828 630
176687-176750
START_RANGE client89a socket2829 620
176750-176812
START_RANGE client89a socket2830 630
176812-176875
START_RANGE client89a socket2831 620
176875-176937
START_RANGE client89a socket2832 630
176937-177000
START_RANGE client89a socket2833 620
177000-177062
START_RANGE client89a socket2834 630
177062-177125
START_RANGE client89a socket2835 620
177125-177187
START_RANGE client89a socket2836 630
177187-177250
START_RANGE client89a socket2837 620
177250-177312
START_RANGE client89a socket2838 630
177312-177375
START_RANGE client89a socket2839 620
177375-177437
START_RANGE client89a socket2840 630
177437-177500
START_RANGE client89a socket2841 620
177500-177562
START_RANGE client89a socket2842 630
177562-177625
START_RANGE client89a socket2843 620
177625-177687
START_RANGE client89a socket2844 630
177687-177750
START_RANGE client89a socket2845 620
177750-177812
START_RANGE client89a socket2846 630
177812-177875
START_RANGE client89a socket2847 620
177875-177937
START_RANGE client89a socket2848 630
177937-178000
START_RANGE client90a socket2849 620
178000-178062
START_RANGE client90a socket2850 630
178062-178125
START_RANGE client90a socket2851 620
178125-178187
START_RANGE client90a socket2852 630
178187-178250
START_RANGE client90a socket2853 620
178250-178312
START_RANGE client90a socket2854 630
178312-178375
START_RANGE client90a socket2855 620
178375-178437
START_RANGE client90a socket2856 630
178437-178500
START_RANGE client90a socket2857 620
178500-178562
START_RANGE client90a socket2858 630
178562-178625
START_RANGE client90a socket2859 620
178625-178687
START_RANGE client90a socket2860 630
178687-178750
START_RANGE client90a socket2861 620
178750-178812
START_RANGE client90a socket2862 630
178812-178875
START_RANGE client90a socket2863 620
178875-178937
START_RANGE client90a socket2864 630
178937-179000
START_RANGE client90a socket2865 620
179000-179062
START_RANGE client90a socket2866 630
179062-179125
START_RANGE client90a socket2867 620
179125-179187
START_RANGE client90a socket2868 630
179187-179250
START_RANGE client90a socket2869 620
179250-179312
START_RANGE client90a socket2870 630
179312-179375
START_RANGE client90a socket2871 620
179375-179437
START_RANGE client90a socket2872 630
179437-179500
START_RANGE client90a socket2873 620
179500-179562
START_RANGE client90a socket2874 630
179562-179625
START_RANGE client90a socket2875 620
179625-179687
START_RANGE client90a socket2876 630
179687-179750
START_RANGE client90a socket2877 620
179750-179812
START_RANGE client90a socket2878 630
179812-179875
START_RANGE client90a socket2879 620
179875-179937
START_RANGE client90a socket2880 630
179937-180000
START_RANGE client91a socket2881 620
180000-180062
START_RANGE client91a socket2882 630
180062-180125
START_RANGE client91a socket2883 620
180125-180187
START_RANGE client91a socket2884 630
180187-180250
START_RANGE client91a socket2885 620
180250-180312
START_RANGE client91a socket2886 630
180312-180375
START_RANGE client91a socket2887 620
180375-180437
START_RANGE client91a socket2888 630
180437-180500
START_RANGE client91a socket2889 620
180500-180562
START_RANGE client91a socket2890 630
180562-180625
START_RANGE client91a socket2891 620
180625-180687
START_RANGE client91a socket2892 630
180687-180750
START_RANGE client91a socket2893 620
180750-180812
START_RANGE client91a socket2894 630
180812-180875
START_RANGE client91a socket2895 620
180875-180937
START_RANGE client91a socket2896 630
180937-181000
START_RANGE client91a socket2897 620
181000-181062
START_RANGE client91a socket2898 630
181062-181125
START_RANGE client91a socket2899 620
181125-181187
START_RANGE client91a socket2900 630
181187-181250
START_RANGE client91a socket2901 620
181250-181312
START_RANGE client91a socket2902 630
181312-181375
START_RANGE client91a socket2903 620
181375-181437
START_RANGE client91a socket2904 630
181437-181500
START_RANGE client91a socket2905 620
181500-181562
START_RANGE client91a socket2906 630
181562-181625
START_RANGE client91a socket2907 620
181625-181687
START_RANGE client91a socket2908 630
181687-181750
START_RANGE client91a socket2909 620
181750-181812
START_RANGE client91a socket2910 630
181812-181875
START_RANGE client91a socket2911 620
181875-181937
START_RANGE client91a socket2912 630
181937-182000
START_RANGE client92a socket2913 620
182000-182062
START_RANGE client92a socket2914 630
182062-182125
START_RANGE client92a socket2915 620
182125-182187
START_RANGE client92a socket2916 630
182187-182250
START_RANGE client92a socket2917 620
182250-182312
START_RANGE client92a socket2918 630
182312-182375
START_RANGE client92a socket2919 620
182375-182437
START_RANGE client92a socket2920 630
182437-182500
START_RANGE client92a socket2921 620
182500-182562
START_RANGE client92a socket2922 630
182562-182625
START_RANGE client92a socket2923 620
182625-182687
START_RANGE client92a socket2924 630
182687-182750
START_RANGE client92a socket2925 620
182750-182812
START_RANGE client92a socket2926 630
182812-182875
START_RANGE client92a socket2927 620
182875-182937
START_RANGE client92a socket2928 630
182937-183000
START_RANGE client92a socket2929 620
183000-183062
START_RANGE client92a socket2930 630
183062-183125
START_RANGE client92a socket2931 620
183125-183187
START_RANGE client92a socket2932 630
183187-183250
START_RANGE client92a socket2933 620
183250-183312
START_RANGE client92a socket2934 630
183312-183375
START_RANGE client92a socket2935 620
183375-183437
START_RANGE client92a socket2936 630
183437-183500
START_RANGE client92a socket2937 620
183500-183562
START_RANGE client92a socket2938 630
183562-183625
START_RANGE client92a socket2939 620
183625-183687
START_RANGE client92a socket2940 630
183687-183750
START_RANGE client92a socket2941 620
183750-183812
START_RANGE client92a socket2942 630
183812-183875
START_RANGE client92a socket2943 620
183875-183937
START_RANGE client92a socket2944 630
183937-184000
#elif MASTER_NUM24
START_RANGE client93a socket2945 620
184000-184062
START_RANGE client93a socket2946 630
184062-184125
```

START_RANGE client93a socket2947 620
184125-184187
START_RANGE client93a socket2948 630
184187-184250
START_RANGE client93a socket2949 620
184250-184312
START_RANGE client93a socket2950 630
184312-184375
START_RANGE client93a socket2951 620
184375-184437
START_RANGE client93a socket2952 630
184437-184500
START_RANGE client93a socket2953 620
184500-184562
START_RANGE client93a socket2954 630
184562-184625
START_RANGE client93a socket2955 620
184625-184687
START_RANGE client93a socket2956 630
184687-184750
START_RANGE client93a socket2957 620
184750-184812
START_RANGE client93a socket2958 630
184812-184875
START_RANGE client93a socket2959 620
184875-184937
START_RANGE client93a socket2960 630
184937-185000
START_RANGE client93a socket2961 620
185000-185062
START_RANGE client93a socket2962 630
185062-185125
START_RANGE client93a socket2963 620
185125-185187
START_RANGE client93a socket2964 630
185187-185250
START_RANGE client93a socket2965 620
185250-185312
START_RANGE client93a socket2966 630
185312-185375
START_RANGE client93a socket2967 620
185375-185437
START_RANGE client93a socket2968 630
185437-185500
START_RANGE client93a socket2969 620
185500-185562
START_RANGE client93a socket2970 630
185562-185625
START_RANGE client93a socket2971 620
185625-185687
START_RANGE client93a socket2972 630
185687-185750
START_RANGE client93a socket2973 620
185750-185812
START_RANGE client93a socket2974 630
185812-185875
START_RANGE client93a socket2975 620
185875-185937
START_RANGE client93a socket2976 630
185937-186000
START_RANGE client94a socket2977 620
186000-186062
START_RANGE client94a socket2978 630
186062-186125
START_RANGE client94a socket2979 620
186125-186187
START_RANGE client94a socket2980 630
186187-186250
START_RANGE client94a socket2981 620
186250-186312
START_RANGE client94a socket2982 630
186312-186375
START_RANGE client94a socket2983 620
186375-186437
START_RANGE client94a socket2984 630
186437-186500
START_RANGE client94a socket2985 620
186500-186562
START_RANGE client94a socket2986 630
186562-186625
START_RANGE client94a socket2987 620
186625-186687
START_RANGE client94a socket2988 630
186687-186750
START_RANGE client94a socket2989 620
186750-186812
START_RANGE client94a socket2990 630
186812-186875
START_RANGE client94a socket2991 620
186875-186937
START_RANGE client94a socket2992 630
186937-187000
START_RANGE client94a socket2993 620
187000-187062
START_RANGE client94a socket2994 630
187062-187125
START_RANGE client94a socket2995 620
187125-187187
START_RANGE client94a socket2996 630
187187-187250
START_RANGE client94a socket2997 620
187250-187312
START_RANGE client94a socket2998 630
187312-187375
START_RANGE client94a socket2999 620
187375-187437
START_RANGE client94a socket3000 630
187437-187500
START_RANGE client94a socket3001 620
187500-187562
START_RANGE client94a socket3002 630
187562-187625
START_RANGE client94a socket3003 620
187625-187687
START_RANGE client94a socket3004 630
187687-187750
START_RANGE client94a socket3005 620
187750-187812
START_RANGE client94a socket3006 630
187812-187875
START_RANGE client94a socket3007 620
187875-187937
START_RANGE client94a socket3008 630
187937-188000
START_RANGE client95a socket3009 620
188000-188062
START_RANGE client95a socket3010 630
188062-188125
START_RANGE client95a socket3011 620
188125-188187
START_RANGE client95a socket3012 630
188187-188250
START_RANGE client95a socket3013 620
188250-188312
START_RANGE client95a socket3014 630
188312-188375
START_RANGE client95a socket3015 620
188375-188437
START_RANGE client95a socket3016 630
188437-188500
START_RANGE client95a socket3017 620
188500-188562
START_RANGE client95a socket3018 630
188562-188625
START_RANGE client95a socket3019 620
188625-188687
START_RANGE client95a socket3020 630
188687-188750
START_RANGE client95a socket3021 620
188750-188812
START_RANGE client95a socket3022 630
188812-188875
START_RANGE client95a socket3023 620
188875-188937
START_RANGE client95a socket3024 630
188937-189000
START_RANGE client95a socket3025 620
189000-189062
START_RANGE client95a socket3026 630
189062-189125
START_RANGE client95a socket3027 620
189125-189187
START_RANGE client95a socket3028 630
189187-189250
START_RANGE client95a socket3029 620
189250-189312
START_RANGE client95a socket3030 630
189312-189375
START_RANGE client95a socket3031 620
189375-189437
START_RANGE client95a socket3032 630
189437-189500
START_RANGE client95a socket3033 620
189500-189562
START_RANGE client95a socket3034 630
189562-189625
START_RANGE client95a socket3035 620
189625-189687
START_RANGE client95a socket3036 630
189687-189750
START_RANGE client95a socket3037 620
189750-189812
START_RANGE client95a socket3038 630
189812-189875
START_RANGE client95a socket3039 620
189875-189937
START_RANGE client95a socket3040 630
189937-190000
START_RANGE client96a socket3041 620
190000-190062
START_RANGE client96a socket3042 630
190062-190125
START_RANGE client96a socket3043 620
190125-190187
START_RANGE client96a socket3044 630
190187-190250
START_RANGE client96a socket3045 620
190250-190312
START_RANGE client96a socket3046 630
190312-190375
START_RANGE client96a socket3047 620
190375-190437
START_RANGE client96a socket3048 630
190437-190500
START_RANGE client96a socket3049 620
190500-190562
START_RANGE client96a socket3050 630
190562-190625
START_RANGE client96a socket3051 620
190625-190687
START_RANGE client96a socket3052 630
190687-190750
START_RANGE client96a socket3053 620
190750-190812
START_RANGE client96a socket3054 630
190812-190875
START_RANGE client96a socket3055 620
190875-190937
START_RANGE client96a socket3056 630
190937-191000
START_RANGE client96a socket3057 620
191000-191062
START_RANGE client96a socket3058 630
191062-191125
START_RANGE client96a socket3059 620
191125-191187
START_RANGE client96a socket3060 630
191187-191250
START_RANGE client96a socket3061 620
191250-191312
START_RANGE client96a socket3062 630
191312-191375
START_RANGE client96a socket3063 620
191375-191437
START_RANGE client96a socket3064 630
191437-191500
START_RANGE client96a socket3065 620
191500-191562
START_RANGE client96a socket3066 630
191562-191625
START_RANGE client96a socket3067 620
191625-191687
START_RANGE client96a socket3068 630
191687-191750
START_RANGE client96a socket3069 620
191750-191812

```
START_RANGE client96a socket3070 630
191812-191875
START_RANGE client96a socket3071 620
191875-191937
START_RANGE client96a socket3072 630
191937-192000
#elif MASTER_NUM25
START_RANGE client97a socket3073 620
192000-192062
START_RANGE client97a socket3074 630
192062-192125
START_RANGE client97a socket3075 620
192125-192187
START_RANGE client97a socket3076 630
192187-192250
START_RANGE client97a socket3077 620
192250-192312
START_RANGE client97a socket3078 630
192312-192375
START_RANGE client97a socket3079 620
192375-192437
START_RANGE client97a socket3080 630
192437-192500
START_RANGE client97a socket3081 620
192500-192562
START_RANGE client97a socket3082 630
192562-192625
START_RANGE client97a socket3083 620
192625-192687
START_RANGE client97a socket3084 630
192687-192750
START_RANGE client97a socket3085 620
192750-192812
START_RANGE client97a socket3086 630
192812-192875
START_RANGE client97a socket3087 620
192875-192937
START_RANGE client97a socket3088 630
192937-193000
START_RANGE client97a socket3089 620
193000-193062
START_RANGE client97a socket3090 630
193062-193125
START_RANGE client97a socket3091 620
193125-193187
START_RANGE client97a socket3092 630
193187-193250
START_RANGE client97a socket3093 620
193250-193312
START_RANGE client97a socket3094 630
193312-193375
START_RANGE client97a socket3095 620
193375-193437
START_RANGE client97a socket3096 630
193437-193500
START_RANGE client97a socket3097 620
193500-193562
START_RANGE client97a socket3098 630
193562-193625
START_RANGE client97a socket3099 620
193625-193687
START_RANGE client97a socket3100 630
193687-193750
START_RANGE client97a socket3101 620
193750-193812
START_RANGE client97a socket3102 630
193812-193875
START_RANGE client97a socket3103 620
193875-193937
START_RANGE client97a socket3104 630
193937-194000
START_RANGE client98a socket3105 620
194000-194062
START_RANGE client98a socket3106 630
194062-194125
START_RANGE client98a socket3107 620
194125-194187
START_RANGE client98a socket3108 630
194187-194250
START_RANGE client98a socket3109 620
194250-194312
START_RANGE client98a socket3110 630
194312-194375
START_RANGE client98a socket3111 620
194375-194437
START_RANGE client98a socket3112 630
194437-194500
START_RANGE client98a socket3113 620
194500-194562
START_RANGE client98a socket3114 630
194562-194625
START_RANGE client98a socket3115 620
194625-194687
START_RANGE client98a socket3116 630
194687-194750
START_RANGE client98a socket3117 620
194750-194812
START_RANGE client98a socket3118 630
194812-194875
START_RANGE client98a socket3119 620
194875-194937
START_RANGE client98a socket3120 630
194937-195000
START_RANGE client98a socket3121 620
195000-195062
START_RANGE client98a socket3122 630
195062-195125
START_RANGE client98a socket3123 620
195125-195187
START_RANGE client98a socket3124 630
195187-195250
START_RANGE client98a socket3125 620
195250-195312
START_RANGE client98a socket3126 630
195312-195375
START_RANGE client98a socket3127 620
195375-195437
START_RANGE client98a socket3128 630
195437-195500
START_RANGE client98a socket3129 620
195500-195562
START_RANGE client98a socket3130 630
195562-195625
START_RANGE client98a socket3131 620
195625-195687
START_RANGE client98a socket3132 630
195687-195750
START_RANGE client98a socket3133 620
195750-195812
START_RANGE client98a socket3134 630
195812-195875
START_RANGE client98a socket3135 620
195875-195937
START_RANGE client98a socket3136 630
195937-196000
START_RANGE client99a socket3137 620
196000-196062
START_RANGE client99a socket3138 630
196062-196125
START_RANGE client99a socket3139 620
196125-196187
START_RANGE client99a socket3140 630
196187-196250
START_RANGE client99a socket3141 620
196250-196312
START_RANGE client99a socket3142 630
196312-196375
START_RANGE client99a socket3143 620
196375-196437
START_RANGE client99a socket3144 630
196437-196500
START_RANGE client99a socket3145 620
196500-196562
START_RANGE client99a socket3146 630
196562-196625
START_RANGE client99a socket3147 620
196625-196687
START_RANGE client99a socket3148 630
196687-196750
START_RANGE client99a socket3149 620
196750-196812
START_RANGE client99a socket3150 630
196812-196875
START_RANGE client99a socket3151 620
196875-196937
START_RANGE client99a socket3152 630
196937-197000
START_RANGE client99a socket3153 620
197000-197062
START_RANGE client99a socket3154 630
197062-197125
START_RANGE client99a socket3155 620
197125-197187
START_RANGE client99a socket3156 630
197187-197250
START_RANGE client99a socket3157 620
197250-197312
START_RANGE client99a socket3158 630
197312-197375
START_RANGE client99a socket3159 620
197375-197437
START_RANGE client99a socket3160 630
197437-197500
START_RANGE client99a socket3161 620
197500-197562
START_RANGE client99a socket3162 630
197562-197625
START_RANGE client99a socket3163 620
197625-197687
START_RANGE client99a socket3164 630
197687-197750
START_RANGE client99a socket3165 620
197750-197812
START_RANGE client99a socket3166 630
197812-197875
START_RANGE client99a socket3167 620
197875-197937
START_RANGE client99a socket3168 630
197937-198000
START_RANGE client100a socket3169 620
198000-198062
START_RANGE client100a socket3170 630
198062-198125
START_RANGE client100a socket3171 620
198125-198187
START_RANGE client100a socket3172 630
198187-198250
START_RANGE client100a socket3173 620
198250-198312
START_RANGE client100a socket3174 630
198312-198375
START_RANGE client100a socket3175 620
198375-198437
START_RANGE client100a socket3176 630
198437-198500
START_RANGE client100a socket3177 620
198500-198562
START_RANGE client100a socket3178 630
198562-198625
START_RANGE client100a socket3179 620
198625-198687
START_RANGE client100a socket3180 630
198687-198750
START_RANGE client100a socket3181 620
198750-198812
START_RANGE client100a socket3182 630
198812-198875
START_RANGE client100a socket3183 620
198875-198937
START_RANGE client100a socket3184 630
198937-199000
START_RANGE client100a socket3185 620
199000-199062
START_RANGE client100a socket3186 630
199062-199125
START_RANGE client100a socket3187 620
199125-199187
START_RANGE client100a socket3188 630
199187-199250
START_RANGE client100a socket3189 620
199250-199312
START_RANGE client100a socket3190 630
199312-199375
START_RANGE client100a socket3191 620
199375-199437
START_RANGE client100a socket3192 630
199437-199500
```

```
START_RANGE client100a socket3193 620
199500-199562
START_RANGE client100a socket3194 630
199562-199625
START_RANGE client100a socket3195 620
199625-199687
START_RANGE client100a socket3196 630
199687-199750
START_RANGE client100a socket3197 620
199750-199812
START_RANGE client100a socket3198 630
199812-199875
START_RANGE client100a socket3199 620
199875-199937
START_RANGE client100a socket3200 630
199937-200000
#elif MASTER_NUM26
START_RANGE client101a socket3201 620
200000-200062
START_RANGE client101a socket3202 630
200062-200125
START_RANGE client101a socket3203 620
200125-200187
START_RANGE client101a socket3204 630
200187-200250
START_RANGE client101a socket3205 620
200250-200312
START_RANGE client101a socket3206 630
200312-200375
START_RANGE client101a socket3207 620
200375-200437
START_RANGE client101a socket3208 630
200437-200500
START_RANGE client101a socket3209 620
200500-200562
START_RANGE client101a socket3210 630
200562-200625
START_RANGE client101a socket3211 620
200625-200687
START_RANGE client101a socket3212 630
200687-200750
START_RANGE client101a socket3213 620
200750-200812
START_RANGE client101a socket3214 630
200812-200875
START_RANGE client101a socket3215 620
200875-200937
START_RANGE client101a socket3216 630
200937-201000
START_RANGE client101a socket3217 620
201000-201062
START_RANGE client101a socket3218 630
201062-201125
START_RANGE client101a socket3219 620
201125-201187
START_RANGE client101a socket3220 630
201187-201250
START_RANGE client101a socket3221 620
201250-201312
START_RANGE client101a socket3222 630
201312-201375
START_RANGE client101a socket3223 620
201375-201437
START_RANGE client101a socket3224 630
201437-201500
START_RANGE client101a socket3225 620
201500-201562
START_RANGE client101a socket3226 630
201562-201625
START_RANGE client101a socket3227 620
201625-201687
START_RANGE client101a socket3228 630
201687-201750
START_RANGE client101a socket3229 620
201750-201812
START_RANGE client101a socket3230 630
201812-201875
START_RANGE client101a socket3231 620
201875-201937
START_RANGE client101a socket3232 630
201937-202000
START_RANGE client102a socket3233 620
202000-202062
START_RANGE client102a socket3234 630
202062-202125
START_RANGE client102a socket3235 620
202125-202187
START_RANGE client102a socket3236 630
202187-202250
START_RANGE client102a socket3237 620
202250-202312
START_RANGE client102a socket3238 630
202312-202375
START_RANGE client102a socket3239 620
202375-202437
START_RANGE client102a socket3240 630
202437-202500
START_RANGE client102a socket3241 620
202500-202562
START_RANGE client102a socket3242 630
202562-202625
START_RANGE client102a socket3243 620
202625-202687
START_RANGE client102a socket3244 630
202687-202750
START_RANGE client102a socket3245 620
202750-202812
START_RANGE client102a socket3246 630
202812-202875
START_RANGE client102a socket3247 620
202875-202937
START_RANGE client102a socket3248 630
202937-203000
START_RANGE client102a socket3249 620
203000-203062
START_RANGE client102a socket3250 630
203062-203125
START_RANGE client102a socket3251 620
203125-203187
START_RANGE client102a socket3252 630
203187-203250
START_RANGE client102a socket3253 620
203250-203312
START_RANGE client102a socket3254 630
203312-203375
START_RANGE client102a socket3255 620
203375-203437
START_RANGE client102a socket3256 630
203437-203500
START_RANGE client102a socket3257 620
203500-203562
START_RANGE client102a socket3258 630
203562-203625
START_RANGE client102a socket3259 620
203625-203687
START_RANGE client102a socket3260 630
203687-203750
START_RANGE client102a socket3261 620
203750-203812
START_RANGE client102a socket3262 630
203812-203875
START_RANGE client102a socket3263 620
203875-203937
START_RANGE client102a socket3264 630
203937-204000
START_RANGE client103a socket3265 620
204000-204062
START_RANGE client103a socket3266 630
204062-204125
START_RANGE client103a socket3267 620
204125-204187
START_RANGE client103a socket3268 630
204187-204250
START_RANGE client103a socket3269 620
204250-204312
START_RANGE client103a socket3270 630
204312-204375
START_RANGE client103a socket3271 620
204375-204437
START_RANGE client103a socket3272 630
204437-204500
START_RANGE client103a socket3273 620
204500-204562
START_RANGE client103a socket3274 630
204562-204625
START_RANGE client103a socket3275 620
204625-204687
START_RANGE client103a socket3276 630
204687-204750
START_RANGE client103a socket3277 620
204750-204812
START_RANGE client103a socket3278 630
204812-204875
START_RANGE client103a socket3279 620
204875-204937
START_RANGE client103a socket3280 630
204937-205000
START_RANGE client103a socket3281 620
205000-205062
START_RANGE client103a socket3282 630
205062-205125
START_RANGE client103a socket3283 620
205125-205187
START_RANGE client103a socket3284 630
205187-205250
START_RANGE client103a socket3285 620
205250-205312
START_RANGE client103a socket3286 630
205312-205375
START_RANGE client103a socket3287 620
205375-205437
START_RANGE client103a socket3288 630
205437-205500
START_RANGE client103a socket3289 620
205500-205562
START_RANGE client103a socket3290 630
205562-205625
START_RANGE client103a socket3291 620
205625-205687
START_RANGE client103a socket3292 630
205687-205750
START_RANGE client103a socket3293 620
205750-205812
START_RANGE client103a socket3294 630
205812-205875
START_RANGE client103a socket3295 620
205875-205937
START_RANGE client103a socket3296 630
205937-206000
START_RANGE client104a socket3297 620
206000-206062
START_RANGE client104a socket3298 630
206062-206125
START_RANGE client104a socket3299 620
206125-206187
START_RANGE client104a socket3300 630
206187-206250
START_RANGE client104a socket3301 620
206250-206312
START_RANGE client104a socket3302 630
206312-206375
START_RANGE client104a socket3303 620
206375-206437
START_RANGE client104a socket3304 630
206437-206500
START_RANGE client104a socket3305 620
206500-206562
START_RANGE client104a socket3306 630
206562-206625
START_RANGE client104a socket3307 620
206625-206687
START_RANGE client104a socket3308 630
206687-206750
START_RANGE client104a socket3309 620
206750-206812
START_RANGE client104a socket3310 630
206812-206875
START_RANGE client104a socket3311 620
206875-206937
START_RANGE client104a socket3312 630
206937-207000
START_RANGE client104a socket3313 620
207000-207062
START_RANGE client104a socket3314 630
207062-207125
START_RANGE client104a socket3315 620
207125-207187
```

START_RANGE client104a socket3316 630 207187-207250
START_RANGE client104a socket3317 620 207250-207312
START_RANGE client104a socket3318 630 207312-207375
START_RANGE client104a socket3319 620 207375-207437
START_RANGE client104a socket3320 630 207437-207500
START_RANGE client104a socket3321 620 207500-207562
START_RANGE client104a socket3322 630 207562-207625
START_RANGE client104a socket3323 620 207625-207687
START_RANGE client104a socket3324 630 207687-207750
START_RANGE client104a socket3325 620 207750-207812
START_RANGE client104a socket3326 630 207812-207875
START_RANGE client104a socket3327 620 207875-207937
START_RANGE client104a socket3328 630 207937-208000
#elif MASTER_NUM27
START_RANGE client105a socket3329 620 208000-208062
START_RANGE client105a socket3330 630 208062-208125
START_RANGE client105a socket3331 620 208125-208187
START_RANGE client105a socket3332 630 208187-208250
START_RANGE client105a socket3333 620 208250-208312
START_RANGE client105a socket3334 630 208312-208375
START_RANGE client105a socket3335 620 208375-208437
START_RANGE client105a socket3336 630 208437-208500
START_RANGE client105a socket3337 620 208500-208562
START_RANGE client105a socket3338 630 208562-208625
START_RANGE client105a socket3339 620 208625-208687
START_RANGE client105a socket3340 630 208687-208750
START_RANGE client105a socket3341 620 208750-208812
START_RANGE client105a socket3342 630 208812-208875
START_RANGE client105a socket3343 620 208875-208937
START_RANGE client105a socket3344 630 208937-209000
START_RANGE client105a socket3345 620 209000-209062
START_RANGE client105a socket3346 630 209062-209125
START_RANGE client105a socket3347 620 209125-209187
START_RANGE client105a socket3348 630 209187-209250
START_RANGE client105a socket3349 620 209250-209312
START_RANGE client105a socket3350 630 209312-209375
START_RANGE client105a socket3351 620 209375-209437
START_RANGE client105a socket3352 630 209437-209500
START_RANGE client105a socket3353 620 209500-209562
START_RANGE client105a socket3354 630 209562-209625
START_RANGE client105a socket3355 620 209625-209687
START_RANGE client105a socket3356 630 209687-209750
START_RANGE client105a socket3357 620 209750-209812
START_RANGE client105a socket3358 630 209812-209875
START_RANGE client105a socket3359 620 209875-209937
START_RANGE client105a socket3360 630 209937-210000
START_RANGE client106a socket3361 620 210000-210062
START_RANGE client106a socket3362 630 210062-210125
START_RANGE client106a socket3363 620 210125-210187
START_RANGE client106a socket3364 630 210187-210250
START_RANGE client106a socket3365 620 210250-210312
START_RANGE client106a socket3366 630 210312-210375
START_RANGE client106a socket3367 620 210375-210437
START_RANGE client106a socket3368 630 210437-210500
START_RANGE client106a socket3369 620 210500-210562
START_RANGE client106a socket3370 630 210562-210625
START_RANGE client106a socket3371 620 210625-210687
START_RANGE client106a socket3372 630 210687-210750
START_RANGE client106a socket3373 620 210750-210812
START_RANGE client106a socket3374 630 210812-210875
START_RANGE client106a socket3375 620 210875-210937
START_RANGE client106a socket3376 630 210937-211000
START_RANGE client106a socket3377 620 211000-211062
START_RANGE client106a socket3378 630 211062-211125
START_RANGE client106a socket3379 620 211125-211187
START_RANGE client106a socket3380 630 211187-211250
START_RANGE client106a socket3381 620 211250-211312
START_RANGE client106a socket3382 630 211312-211375
START_RANGE client106a socket3383 620 211375-211437
START_RANGE client106a socket3384 630 211437-211500
START_RANGE client106a socket3385 620 211500-211562
START_RANGE client106a socket3386 630 211562-211625
START_RANGE client106a socket3387 620 211625-211687
START_RANGE client106a socket3388 630 211687-211750
START_RANGE client106a socket3389 620 211750-211812
START_RANGE client106a socket3390 630 211812-211875
START_RANGE client106a socket3391 620 211875-211937
START_RANGE client106a socket3392 630 211937-212000
START_RANGE client107a socket3393 620 212000-212062
START_RANGE client107a socket3394 630 212062-212125
START_RANGE client107a socket3395 620 212125-212187
START_RANGE client107a socket3396 630 212187-212250
START_RANGE client107a socket3397 620 212250-212312
START_RANGE client107a socket3398 630 212312-212375
START_RANGE client107a socket3399 620 212375-212437
START_RANGE client107a socket3400 630 212437-212500
START_RANGE client107a socket3401 620 212500-212562
START_RANGE client107a socket3402 630 212562-212625
START_RANGE client107a socket3403 620 212625-212687
START_RANGE client107a socket3404 630 212687-212750
START_RANGE client107a socket3405 620 212750-212812
START_RANGE client107a socket3406 630 212812-212875
START_RANGE client107a socket3407 620 212875-212937
START_RANGE client107a socket3408 630 212937-213000
START_RANGE client107a socket3409 620 213000-213062
START_RANGE client107a socket3410 630 213062-213125
START_RANGE client107a socket3411 620 213125-213187
START_RANGE client107a socket3412 630 213187-213250
START_RANGE client107a socket3413 620 213250-213312
START_RANGE client107a socket3414 630 213312-213375
START_RANGE client107a socket3415 620 213375-213437
START_RANGE client107a socket3416 630 213437-213500
START_RANGE client107a socket3417 620 213500-213562
START_RANGE client107a socket3418 630 213562-213625
START_RANGE client107a socket3419 620 213625-213687
START_RANGE client107a socket3420 630 213687-213750
START_RANGE client107a socket3421 620 213750-213812
START_RANGE client107a socket3422 630 213812-213875
START_RANGE client107a socket3423 620 213875-213937
START_RANGE client107a socket3424 630 213937-214000
START_RANGE client108a socket3425 620 214000-214062
START_RANGE client108a socket3426 630 214062-214125
START_RANGE client108a socket3427 620 214125-214187
START_RANGE client108a socket3428 630 214187-214250
START_RANGE client108a socket3429 620 214250-214312
START_RANGE client108a socket3430 630 214312-214375
START_RANGE client108a socket3431 620 214375-214437
START_RANGE client108a socket3432 630 214437-214500
START_RANGE client108a socket3433 620 214500-214562
START_RANGE client108a socket3434 630 214562-214625
START_RANGE client108a socket3435 620 214625-214687
START_RANGE client108a socket3436 630 214687-214750
START_RANGE client108a socket3437 620 214750-214812
START_RANGE client108a socket3438 630 214812-214875

```
START_RANGE client108a socket3439 620
214875-214937
START_RANGE client108a socket3440 630
214937-215000
START_RANGE client108a socket3441 620
215000-215062
START_RANGE client108a socket3442 630
215062-215125
START_RANGE client108a socket3443 620
215125-215187
START_RANGE client108a socket3444 630
215187-215250
START_RANGE client108a socket3445 620
215250-215312
START_RANGE client108a socket3446 630
215312-215375
START_RANGE client108a socket3447 620
215375-215437
START_RANGE client108a socket3448 630
215437-215500
START_RANGE client108a socket3449 620
215500-215562
START_RANGE client108a socket3450 630
215562-215625
START_RANGE client108a socket3451 620
215625-215687
START_RANGE client108a socket3452 630
215687-215750
START_RANGE client108a socket3453 620
215750-215812
START_RANGE client108a socket3454 630
215812-215875
START_RANGE client108a socket3455 620
215875-215937
START_RANGE client108a socket3456 630
215937-216000
#elif MASTER_NUM28
START_RANGE client109a socket3457 620
216000-216062
START_RANGE client109a socket3458 630
216062-216125
START_RANGE client109a socket3459 620
216125-216187
START_RANGE client109a socket3460 630
216187-216250
START_RANGE client109a socket3461 620
216250-216312
START_RANGE client109a socket3462 630
216312-216375
START_RANGE client109a socket3463 620
216375-216437
START_RANGE client109a socket3464 630
216437-216500
START_RANGE client109a socket3465 620
216500-216562
START_RANGE client109a socket3466 630
216562-216625
START_RANGE client109a socket3467 620
216625-216687
START_RANGE client109a socket3468 630
216687-216750
START_RANGE client109a socket3469 620
216750-216812
START_RANGE client109a socket3470 630
216812-216875
START_RANGE client109a socket3471 620
216875-216937
START_RANGE client109a socket3472 630
216937-217000
START_RANGE client109a socket3473 620
217000-217062
START_RANGE client109a socket3474 630
217062-217125
START_RANGE client109a socket3475 620
217125-217187
START_RANGE client109a socket3476 630
217187-217250
START_RANGE client109a socket3477 620
217250-217312
START_RANGE client109a socket3478 630
217312-217375
START_RANGE client109a socket3479 620
217375-217437
START_RANGE client109a socket3480 630
217437-217500
START_RANGE client109a socket3481 620
217500-217562
START_RANGE client109a socket3482 630
217562-217625
START_RANGE client109a socket3483 620
217625-217687
START_RANGE client109a socket3484 630
217687-217750
START_RANGE client109a socket3485 620
217750-217812
START_RANGE client109a socket3486 630
217812-217875
START_RANGE client109a socket3487 620
217875-217937
START_RANGE client109a socket3488 630
217937-218000
START_RANGE client110a socket3489 620
218000-218062
START_RANGE client110a socket3490 630
218062-218125
START_RANGE client110a socket3491 620
218125-218187
START_RANGE client110a socket3492 630
218187-218250
START_RANGE client110a socket3493 620
218250-218312
START_RANGE client110a socket3494 630
218312-218375
START_RANGE client110a socket3495 620
218375-218437
START_RANGE client110a socket3496 630
218437-218500
START_RANGE client110a socket3497 620
218500-218562
START_RANGE client110a socket3498 630
218562-218625
START_RANGE client110a socket3499 620
218625-218687
START_RANGE client110a socket3500 630
218687-218750
START_RANGE client110a socket3501 620
218750-218812
START_RANGE client110a socket3502 630
218812-218875
START_RANGE client110a socket3503 620
218875-218937
START_RANGE client110a socket3504 630
218937-219000
START_RANGE client110a socket3505 620
219000-219062
START_RANGE client110a socket3506 630
219062-219125
START_RANGE client110a socket3507 620
219125-219187
START_RANGE client110a socket3508 630
219187-219250
START_RANGE client110a socket3509 620
219250-219312
START_RANGE client110a socket3510 630
219312-219375
START_RANGE client110a socket3511 620
219375-219437
START_RANGE client110a socket3512 630
219437-219500
START_RANGE client110a socket3513 620
219500-219562
START_RANGE client110a socket3514 630
219562-219625
START_RANGE client110a socket3515 620
219625-219687
START_RANGE client110a socket3516 630
219687-219750
START_RANGE client110a socket3517 620
219750-219812
START_RANGE client110a socket3518 630
219812-219875
START_RANGE client110a socket3519 620
219875-219937
START_RANGE client110a socket3520 630
219937-220000
START_RANGE client111a socket3521 620
220000-220062
START_RANGE client111a socket3522 630
220062-220125
START_RANGE client111a socket3523 620
220125-220187
START_RANGE client111a socket3524 630
220187-220250
START_RANGE client111a socket3525 620
220250-220312
START_RANGE client111a socket3526 630
220312-220375
START_RANGE client111a socket3527 620
220375-220437
START_RANGE client111a socket3528 630
220437-220500
START_RANGE client111a socket3529 620
220500-220562
START_RANGE client111a socket3530 630
220562-220625
START_RANGE client111a socket3531 620
220625-220687
START_RANGE client111a socket3532 630
220687-220750
START_RANGE client111a socket3533 620
220750-220812
START_RANGE client111a socket3534 630
220812-220875
START_RANGE client111a socket3535 620
220875-220937
START_RANGE client111a socket3536 630
220937-221000
START_RANGE client111a socket3537 620
221000-221062
START_RANGE client111a socket3538 630
221062-221125
START_RANGE client111a socket3539 620
221125-221187
START_RANGE client111a socket3540 630
221187-221250
START_RANGE client111a socket3541 620
221250-221312
START_RANGE client111a socket3542 630
221312-221375
START_RANGE client111a socket3543 620
221375-221437
START_RANGE client111a socket3544 630
221437-221500
START_RANGE client111a socket3545 620
221500-221562
START_RANGE client111a socket3546 630
221562-221625
START_RANGE client111a socket3547 620
221625-221687
START_RANGE client111a socket3548 630
221687-221750
START_RANGE client111a socket3549 620
221750-221812
START_RANGE client111a socket3550 630
221812-221875
START_RANGE client111a socket3551 620
221875-221937
START_RANGE client111a socket3552 630
221937-222000
START_RANGE client112a socket3553 620
222000-222062
START_RANGE client112a socket3554 630
222062-222125
START_RANGE client112a socket3555 620
222125-222187
START_RANGE client112a socket3556 630
222187-222250
START_RANGE client112a socket3557 620
222250-222312
START_RANGE client112a socket3558 630
222312-222375
START_RANGE client112a socket3559 620
222375-222437
START_RANGE client112a socket3560 630
222437-222500
START_RANGE client112a socket3561 620
222500-222562
```

```
START_RANGE client112a socket3562 630
222562-222625
START_RANGE client112a socket3563 620
222625-222687
START_RANGE client112a socket3564 630
222687-222750
START_RANGE client112a socket3565 620
222750-222812
START_RANGE client112a socket3566 630
222812-222875
START_RANGE client112a socket3567 620
222875-222937
START_RANGE client112a socket3568 630
222937-223000
START_RANGE client112a socket3569 620
223000-223062
START_RANGE client112a socket3570 630
223062-223125
START_RANGE client112a socket3571 620
223125-223187
START_RANGE client112a socket3572 630
223187-223250
START_RANGE client112a socket3573 620
223250-223312
START_RANGE client112a socket3574 630
223312-223375
START_RANGE client112a socket3575 620
223375-223437
START_RANGE client112a socket3576 630
223437-223500
START_RANGE client112a socket3577 620
223500-223562
START_RANGE client112a socket3578 630
223562-223625
START_RANGE client112a socket3579 620
223625-223687
START_RANGE client112a socket3580 630
223687-223750
START_RANGE client112a socket3581 620
223750-223812
START_RANGE client112a socket3582 630
223812-223875
START_RANGE client112a socket3583 620
223875-223937
START_RANGE client112a socket3584 630
223937-224000
#elif MASTER_NUM29
START_RANGE client113a socket3585 620
224000-224062
START_RANGE client113a socket3586 630
224062-224125
START_RANGE client113a socket3587 620
224125-224187
START_RANGE client113a socket3588 630
224187-224250
START_RANGE client113a socket3589 620
224250-224312
START_RANGE client113a socket3590 630
224312-224375
START_RANGE client113a socket3591 620
224375-224437
START_RANGE client113a socket3592 630
224437-224500
START_RANGE client113a socket3593 620
224500-224562
START_RANGE client113a socket3594 630
224562-224625
START_RANGE client113a socket3595 620
224625-224687
START_RANGE client113a socket3596 630
224687-224750
START_RANGE client113a socket3597 620
224750-224812
START_RANGE client113a socket3598 630
224812-224875
START_RANGE client113a socket3599 620
224875-224937
START_RANGE client113a socket3600 630
224937-225000
START_RANGE client113a socket3601 620
225000-225062
START_RANGE client113a socket3602 630
225062-225125
START_RANGE client113a socket3603 620
225125-225187
START_RANGE client113a socket3604 630
225187-225250
START_RANGE client113a socket3605 620
225250-225312
START_RANGE client113a socket3606 630
225312-225375
START_RANGE client113a socket3607 620
225375-225437
START_RANGE client113a socket3608 630
225437-225500
START_RANGE client113a socket3609 620
225500-225562
START_RANGE client113a socket3610 630
225562-225625
START_RANGE client113a socket3611 620
225625-225687
START_RANGE client113a socket3612 630
225687-225750
START_RANGE client113a socket3613 620
225750-225812
START_RANGE client113a socket3614 630
225812-225875
START_RANGE client113a socket3615 620
225875-225937
START_RANGE client113a socket3616 630
225937-226000
START_RANGE client114a socket3617 620
226000-226062
START_RANGE client114a socket3618 630
226062-226125
START_RANGE client114a socket3619 620
226125-226187
START_RANGE client114a socket3620 630
226187-226250
START_RANGE client114a socket3621 620
226250-226312
START_RANGE client114a socket3622 630
226312-226375
START_RANGE client114a socket3623 620
226375-226437
START_RANGE client114a socket3624 630
226437-226500
START_RANGE client114a socket3625 620
226500-226562
START_RANGE client114a socket3626 630
226562-226625
START_RANGE client114a socket3627 620
226625-226687
START_RANGE client114a socket3628 630
226687-226750
START_RANGE client114a socket3629 620
226750-226812
START_RANGE client114a socket3630 630
226812-226875
START_RANGE client114a socket3631 620
226875-226937
START_RANGE client114a socket3632 630
226937-227000
START_RANGE client114a socket3633 620
227000-227062
START_RANGE client114a socket3634 630
227062-227125
START_RANGE client114a socket3635 620
227125-227187
START_RANGE client114a socket3636 630
227187-227250
START_RANGE client114a socket3637 620
227250-227312
START_RANGE client114a socket3638 630
227312-227375
START_RANGE client114a socket3639 620
227375-227437
START_RANGE client114a socket3640 630
227437-227500
START_RANGE client114a socket3641 620
227500-227562
START_RANGE client114a socket3642 630
227562-227625
START_RANGE client114a socket3643 620
227625-227687
START_RANGE client114a socket3644 630
227687-227750
START_RANGE client114a socket3645 620
227750-227812
START_RANGE client114a socket3646 630
227812-227875
START_RANGE client114a socket3647 620
227875-227937
START_RANGE client114a socket3648 630
227937-228000
START_RANGE client115a socket3649 620
228000-228062
START_RANGE client115a socket3650 630
228062-228125
START_RANGE client115a socket3651 620
228125-228187
START_RANGE client115a socket3652 630
228187-228250
START_RANGE client115a socket3653 620
228250-228312
START_RANGE client115a socket3654 630
228312-228375
START_RANGE client115a socket3655 620
228375-228437
START_RANGE client115a socket3656 630
228437-228500
START_RANGE client115a socket3657 620
228500-228562
START_RANGE client115a socket3658 630
228562-228625
START_RANGE client115a socket3659 620
228625-228687
START_RANGE client115a socket3660 630
228687-228750
START_RANGE client115a socket3661 620
228750-228812
START_RANGE client115a socket3662 630
228812-228875
START_RANGE client115a socket3663 620
228875-228937
START_RANGE client115a socket3664 630
228937-229000
START_RANGE client115a socket3665 620
229000-229062
START_RANGE client115a socket3666 630
229062-229125
START_RANGE client115a socket3667 620
229125-229187
START_RANGE client115a socket3668 630
229187-229250
START_RANGE client115a socket3669 620
229250-229312
START_RANGE client115a socket3670 630
229312-229375
START_RANGE client115a socket3671 620
229375-229437
START_RANGE client115a socket3672 630
229437-229500
START_RANGE client115a socket3673 620
229500-229562
START_RANGE client115a socket3674 630
229562-229625
START_RANGE client115a socket3675 620
229625-229687
START_RANGE client115a socket3676 630
229687-229750
START_RANGE client115a socket3677 620
229750-229812
START_RANGE client115a socket3678 630
229812-229875
START_RANGE client115a socket3679 620
229875-229937
START_RANGE client115a socket3680 630
229937-230000
START_RANGE client116a socket3681 620
230000-230062
START_RANGE client116a socket3682 630
230062-230125
START_RANGE client116a socket3683 620
230125-230187
START_RANGE client116a socket3684 630
230187-230250
```

```
START_RANGE client116a socket3685 620
230250-230312
START_RANGE client116a socket3686 630
230312-230375
START_RANGE client116a socket3687 620
230375-230437
START_RANGE client116a socket3688 630
230437-230500
START_RANGE client116a socket3689 620
230500-230562
START_RANGE client116a socket3690 630
230562-230625
START_RANGE client116a socket3691 620
230625-230687
START_RANGE client116a socket3692 630
230687-230750
START_RANGE client116a socket3693 620
230750-230812
START_RANGE client116a socket3694 630
230812-230875
START_RANGE client116a socket3695 620
230875-230937
START_RANGE client116a socket3696 630
230937-231000
START_RANGE client116a socket3697 620
231000-231062
START_RANGE client116a socket3698 630
231062-231125
START_RANGE client116a socket3699 620
231125-231187
START_RANGE client116a socket3700 630
231187-231250
START_RANGE client116a socket3701 620
231250-231312
START_RANGE client116a socket3702 630
231312-231375
START_RANGE client116a socket3703 620
231375-231437
START_RANGE client116a socket3704 630
231437-231500
START_RANGE client116a socket3705 620
231500-231562
START_RANGE client116a socket3706 630
231562-231625
START_RANGE client116a socket3707 620
231625-231687
START_RANGE client116a socket3708 630
231687-231750
START_RANGE client116a socket3709 620
231750-231812
START_RANGE client116a socket3710 630
231812-231875
START_RANGE client116a socket3711 620
231875-231937
START_RANGE client116a socket3712 630
231937-232000
#elif MASTER_NUM30
START_RANGE client117a socket3713 620
232000-232062
START_RANGE client117a socket3714 630
232062-232125
START_RANGE client117a socket3715 620
232125-232187
START_RANGE client117a socket3716 630
232187-232250
START_RANGE client117a socket3717 620
232250-232312
START_RANGE client117a socket3718 630
232312-232375
START_RANGE client117a socket3719 620
232375-232437
START_RANGE client117a socket3720 630
232437-232500
START_RANGE client117a socket3721 620
232500-232562
START_RANGE client117a socket3722 630
232562-232625
START_RANGE client117a socket3723 620
232625-232687
START_RANGE client117a socket3724 630
232687-232750
START_RANGE client117a socket3725 620
232750-232812
START_RANGE client117a socket3726 630
232812-232875
START_RANGE client117a socket3727 620
232875-232937
START_RANGE client117a socket3728 630
232937-233000
START_RANGE client117a socket3729 620
233000-233062
START_RANGE client117a socket3730 630
233062-233125
START_RANGE client117a socket3731 620
233125-233187
START_RANGE client117a socket3732 630
233187-233250
START_RANGE client117a socket3733 620
233250-233312
START_RANGE client117a socket3734 630
233312-233375
START_RANGE client117a socket3735 620
233375-233437
START_RANGE client117a socket3736 630
233437-233500
START_RANGE client117a socket3737 620
233500-233562
START_RANGE client117a socket3738 630
233562-233625
START_RANGE client117a socket3739 620
233625-233687
START_RANGE client117a socket3740 630
233687-233750
START_RANGE client117a socket3741 620
233750-233812
START_RANGE client117a socket3742 630
233812-233875
START_RANGE client117a socket3743 620
233875-233937
START_RANGE client117a socket3744 630
233937-234000
START_RANGE client118a socket3745 620
234000-234062
START_RANGE client118a socket3746 630
234062-234125
START_RANGE client118a socket3747 620
234125-234187
START_RANGE client118a socket3748 630
234187-234250
START_RANGE client118a socket3749 620
234250-234312
START_RANGE client118a socket3750 630
234312-234375
START_RANGE client118a socket3751 620
234375-234437
START_RANGE client118a socket3752 630
234437-234500
START_RANGE client118a socket3753 620
234500-234562
START_RANGE client118a socket3754 630
234562-234625
START_RANGE client118a socket3755 620
234625-234687
START_RANGE client118a socket3756 630
234687-234750
START_RANGE client118a socket3757 620
234750-234812
START_RANGE client118a socket3758 630
234812-234875
START_RANGE client118a socket3759 620
234875-234937
START_RANGE client118a socket3760 630
234937-235000
START_RANGE client118a socket3761 620
235000-235062
START_RANGE client118a socket3762 630
235062-235125
START_RANGE client118a socket3763 620
235125-235187
START_RANGE client118a socket3764 630
235187-235250
START_RANGE client118a socket3765 620
235250-235312
START_RANGE client118a socket3766 630
235312-235375
START_RANGE client118a socket3767 620
235375-235437
START_RANGE client118a socket3768 630
235437-235500
START_RANGE client118a socket3769 620
235500-235562
START_RANGE client118a socket3770 630
235562-235625
START_RANGE client118a socket3771 620
235625-235687
START_RANGE client118a socket3772 630
235687-235750
START_RANGE client118a socket3773 620
235750-235812
START_RANGE client118a socket3774 630
235812-235875
START_RANGE client118a socket3775 620
235875-235937
START_RANGE client118a socket3776 630
235937-236000
START_RANGE client119a socket3777 620
236000-236062
START_RANGE client119a socket3778 630
236062-236125
START_RANGE client119a socket3779 620
236125-236187
START_RANGE client119a socket3780 630
236187-236250
START_RANGE client119a socket3781 620
236250-236312
START_RANGE client119a socket3782 630
236312-236375
START_RANGE client119a socket3783 620
236375-236437
START_RANGE client119a socket3784 630
236437-236500
START_RANGE client119a socket3785 620
236500-236562
START_RANGE client119a socket3786 630
236562-236625
START_RANGE client119a socket3787 620
236625-236687
START_RANGE client119a socket3788 630
236687-236750
START_RANGE client119a socket3789 620
236750-236812
START_RANGE client119a socket3790 630
236812-236875
START_RANGE client119a socket3791 620
236875-236937
START_RANGE client119a socket3792 630
236937-237000
START_RANGE client119a socket3793 620
237000-237062
START_RANGE client119a socket3794 630
237062-237125
START_RANGE client119a socket3795 620
237125-237187
START_RANGE client119a socket3796 630
237187-237250
START_RANGE client119a socket3797 620
237250-237312
START_RANGE client119a socket3798 630
237312-237375
START_RANGE client119a socket3799 620
237375-237437
START_RANGE client119a socket3800 630
237437-237500
START_RANGE client119a socket3801 620
237500-237562
START_RANGE client119a socket3802 630
237562-237625
START_RANGE client119a socket3803 620
237625-237687
START_RANGE client119a socket3804 630
237687-237750
START_RANGE client119a socket3805 620
237750-237812
START_RANGE client119a socket3806 630
237812-237875
START_RANGE client119a socket3807 620
237875-237937
```

```
START_RANGE client119a socket3808 630
237937-238000
START_RANGE client120a socket3809 620
238000-238062
START_RANGE client120a socket3810 630
238062-238125
START_RANGE client120a socket3811 620
238125-238187
START_RANGE client120a socket3812 630
238187-238250
START_RANGE client120a socket3813 620
238250-238312
START_RANGE client120a socket3814 630
238312-238375
START_RANGE client120a socket3815 620
238375-238437
START_RANGE client120a socket3816 630
238437-238500
START_RANGE client120a socket3817 620
238500-238562
START_RANGE client120a socket3818 630
238562-238625
START_RANGE client120a socket3819 620
238625-238687
START_RANGE client120a socket3820 630
238687-238750
START_RANGE client120a socket3821 620
238750-238812
START_RANGE client120a socket3822 630
238812-238875
START_RANGE client120a socket3823 620
238875-238937
START_RANGE client120a socket3824 630
238937-239000
START_RANGE client120a socket3825 620
239000-239062
START_RANGE client120a socket3826 630
239062-239125
START_RANGE client120a socket3827 620
239125-239187
START_RANGE client120a socket3828 630
239187-239250
START_RANGE client120a socket3829 620
239250-239312
START_RANGE client120a socket3830 630
239312-239375
START_RANGE client120a socket3831 620
239375-239437
START_RANGE client120a socket3832 630
239437-239500
START_RANGE client120a socket3833 620
239500-239562
START_RANGE client120a socket3834 630
239562-239625
START_RANGE client120a socket3835 620
239625-239687
START_RANGE client120a socket3836 630
239687-239750
START_RANGE client120a socket3837 620
239750-239812
START_RANGE client120a socket3838 630
239812-239875
START_RANGE client120a socket3839 620
239875-239937
START_RANGE client120a socket3840 630
239937-240000
#elif MASTER_NUM31
START_RANGE client121a socket3841 620
240000-240062
START_RANGE client121a socket3842 630
240062-240125
START_RANGE client121a socket3843 620
240125-240187
START_RANGE client121a socket3844 630
240187-240250
START_RANGE client121a socket3845 620
240250-240312
START_RANGE client121a socket3846 630
240312-240375
START_RANGE client121a socket3847 620
240375-240437
START_RANGE client121a socket3848 630
240437-240500
START_RANGE client121a socket3849 620
240500-240562
START_RANGE client121a socket3850 630
240562-240625
START_RANGE client121a socket3851 620
240625-240687
START_RANGE client121a socket3852 630
240687-240750
START_RANGE client121a socket3853 620
240750-240812
START_RANGE client121a socket3854 630
240812-240875
START_RANGE client121a socket3855 620
240875-240937
START_RANGE client121a socket3856 630
240937-241000
START_RANGE client121a socket3857 620
241000-241062
START_RANGE client121a socket3858 630
241062-241125
START_RANGE client121a socket3859 620
241125-241187
START_RANGE client121a socket3860 630
241187-241250
START_RANGE client121a socket3861 620
241250-241312
START_RANGE client121a socket3862 630
241312-241375
START_RANGE client121a socket3863 620
241375-241437
START_RANGE client121a socket3864 630
241437-241500
START_RANGE client121a socket3865 620
241500-241562
START_RANGE client121a socket3866 630
241562-241625
START_RANGE client121a socket3867 620
241625-241687
START_RANGE client121a socket3868 630
241687-241750
START_RANGE client121a socket3869 620
241750-241812
START_RANGE client121a socket3870 630
241812-241875
START_RANGE client121a socket3871 620
241875-241937
START_RANGE client121a socket3872 630
241937-242000
START_RANGE client122a socket3873 620
242000-242062
START_RANGE client122a socket3874 630
242062-242125
START_RANGE client122a socket3875 620
242125-242187
START_RANGE client122a socket3876 630
242187-242250
START_RANGE client122a socket3877 620
242250-242312
START_RANGE client122a socket3878 630
242312-242375
START_RANGE client122a socket3879 620
242375-242437
START_RANGE client122a socket3880 630
242437-242500
START_RANGE client122a socket3881 620
242500-242562
START_RANGE client122a socket3882 630
242562-242625
START_RANGE client122a socket3883 620
242625-242687
START_RANGE client122a socket3884 630
242687-242750
START_RANGE client122a socket3885 620
242750-242812
START_RANGE client122a socket3886 630
242812-242875
START_RANGE client122a socket3887 620
242875-242937
START_RANGE client122a socket3888 630
242937-243000
START_RANGE client122a socket3889 620
243000-243062
START_RANGE client122a socket3890 630
243062-243125
START_RANGE client122a socket3891 620
243125-243187
START_RANGE client122a socket3892 630
243187-243250
START_RANGE client122a socket3893 620
243250-243312
START_RANGE client122a socket3894 630
243312-243375
START_RANGE client122a socket3895 620
243375-243437
START_RANGE client122a socket3896 630
243437-243500
START_RANGE client122a socket3897 620
243500-243562
START_RANGE client122a socket3898 630
243562-243625
START_RANGE client122a socket3899 620
243625-243687
START_RANGE client122a socket3900 630
243687-243750
START_RANGE client122a socket3901 620
243750-243812
START_RANGE client122a socket3902 630
243812-243875
START_RANGE client122a socket3903 620
243875-243937
START_RANGE client122a socket3904 630
243937-244000
START_RANGE client123a socket3905 620
244000-244062
START_RANGE client123a socket3906 630
244062-244125
START_RANGE client123a socket3907 620
244125-244187
START_RANGE client123a socket3908 630
244187-244250
START_RANGE client123a socket3909 620
244250-244312
START_RANGE client123a socket3910 630
244312-244375
START_RANGE client123a socket3911 620
244375-244437
START_RANGE client123a socket3912 630
244437-244500
START_RANGE client123a socket3913 620
244500-244562
START_RANGE client123a socket3914 630
244562-244625
START_RANGE client123a socket3915 620
244625-244687
START_RANGE client123a socket3916 630
244687-244750
START_RANGE client123a socket3917 620
244750-244812
START_RANGE client123a socket3918 630
244812-244875
START_RANGE client123a socket3919 620
244875-244937
START_RANGE client123a socket3920 630
244937-245000
START_RANGE client123a socket3921 620
245000-245062
START_RANGE client123a socket3922 630
245062-245125
START_RANGE client123a socket3923 620
245125-245187
START_RANGE client123a socket3924 630
245187-245250
START_RANGE client123a socket3925 620
245250-245312
START_RANGE client123a socket3926 630
245312-245375
START_RANGE client123a socket3927 620
245375-245437
START_RANGE client123a socket3928 630
245437-245500
START_RANGE client123a socket3929 620
245500-245562
START_RANGE client123a socket3930 630
245562-245625
```

```
START_RANGE client123a socket3931 620
245625-245687
START_RANGE client123a socket3932 630
245687-245750
START_RANGE client123a socket3933 620
245750-245812
START_RANGE client123a socket3934 630
245812-245875
START_RANGE client123a socket3935 620
245875-245937
START_RANGE client123a socket3936 630
245937-246000
START_RANGE client124a socket3937 620
246000-246062
START_RANGE client124a socket3938 630
246062-246125
START_RANGE client124a socket3939 620
246125-246187
START_RANGE client124a socket3940 630
246187-246250
START_RANGE client124a socket3941 620
246250-246312
START_RANGE client124a socket3942 630
246312-246375
START_RANGE client124a socket3943 620
246375-246437
START_RANGE client124a socket3944 630
246437-246500
START_RANGE client124a socket3945 620
246500-246562
START_RANGE client124a socket3946 630
246562-246625
START_RANGE client124a socket3947 620
246625-246687
START_RANGE client124a socket3948 630
246687-246750
START_RANGE client124a socket3949 620
246750-246812
START_RANGE client124a socket3950 630
246812-246875
START_RANGE client124a socket3951 620
246875-246937
START_RANGE client124a socket3952 630
246937-247000
START_RANGE client124a socket3953 620
247000-247062
START_RANGE client124a socket3954 630
247062-247125
START_RANGE client124a socket3955 620
247125-247187
START_RANGE client124a socket3956 630
247187-247250
START_RANGE client124a socket3957 620
247250-247312
START_RANGE client124a socket3958 630
247312-247375
START_RANGE client124a socket3959 620
247375-247437
START_RANGE client124a socket3960 630
247437-247500
START_RANGE client124a socket3961 620
247500-247562
START_RANGE client124a socket3962 630
247562-247625
START_RANGE client124a socket3963 620
247625-247687
START_RANGE client124a socket3964 630
247687-247750
START_RANGE client124a socket3965 620
247750-247812
START_RANGE client124a socket3966 630
247812-247875
START_RANGE client124a socket3967 620
247875-247937
START_RANGE client124a socket3968 630
247937-248000
#elif MASTER_NUM32
START_RANGE client125a socket3969 620
248000-248062
START_RANGE client125a socket3970 630
248062-248125
START_RANGE client125a socket3971 620
248125-248187
START_RANGE client125a socket3972 630
248187-248250
START_RANGE client125a socket3973 620
248250-248312
START_RANGE client125a socket3974 630
248312-248375
START_RANGE client125a socket3975 620
248375-248437
START_RANGE client125a socket3976 630
248437-248500
START_RANGE client125a socket3977 620
248500-248562
START_RANGE client125a socket3978 630
248562-248625
START_RANGE client125a socket3979 620
248625-248687
START_RANGE client125a socket3980 630
248687-248750
START_RANGE client125a socket3981 620
248750-248812
START_RANGE client125a socket3982 630
248812-248875
START_RANGE client125a socket3983 620
248875-248937
START_RANGE client125a socket3984 630
248937-249000
START_RANGE client125a socket3985 620
249000-249062
START_RANGE client125a socket3986 630
249062-249125
START_RANGE client125a socket3987 620
249125-249187
START_RANGE client125a socket3988 630
249187-249250
START_RANGE client125a socket3989 620
249250-249312
START_RANGE client125a socket3990 630
249312-249375
START_RANGE client125a socket3991 620
249375-249437
START_RANGE client125a socket3992 630
249437-249500
START_RANGE client125a socket3993 620
249500-249562
START_RANGE client125a socket3994 630
249562-249625
START_RANGE client125a socket3995 620
249625-249687
START_RANGE client125a socket3996 630
249687-249750
START_RANGE client125a socket3997 620
249750-249812
START_RANGE client125a socket3998 630
249812-249875
START_RANGE client125a socket3999 620
249875-249937
START_RANGE client125a socket4000 630
249937-250000
START_RANGE client126a socket4001 620
250000-250062
START_RANGE client126a socket4002 630
250062-250125
START_RANGE client126a socket4003 620
250125-250187
START_RANGE client126a socket4004 630
250187-250250
START_RANGE client126a socket4005 620
250250-250312
START_RANGE client126a socket4006 630
250312-250375
START_RANGE client126a socket4007 620
250375-250437
START_RANGE client126a socket4008 630
250437-250500
START_RANGE client126a socket4009 620
250500-250562
START_RANGE client126a socket4010 630
250562-250625
START_RANGE client126a socket4011 620
250625-250687
START_RANGE client126a socket4012 630
250687-250750
START_RANGE client126a socket4013 620
250750-250812
START_RANGE client126a socket4014 630
250812-250875
START_RANGE client126a socket4015 620
250875-250937
START_RANGE client126a socket4016 630
250937-251000
START_RANGE client126a socket4017 620
251000-251062
START_RANGE client126a socket4018 630
251062-251125
START_RANGE client126a socket4019 620
251125-251187
START_RANGE client126a socket4020 630
251187-251250
START_RANGE client126a socket4021 620
251250-251312
START_RANGE client126a socket4022 630
251312-251375
START_RANGE client126a socket4023 620
251375-251437
START_RANGE client126a socket4024 630
251437-251500
START_RANGE client126a socket4025 620
251500-251562
START_RANGE client126a socket4026 630
251562-251625
START_RANGE client126a socket4027 620
251625-251687
START_RANGE client126a socket4028 630
251687-251750
START_RANGE client126a socket4029 620
251750-251812
START_RANGE client126a socket4030 630
251812-251875
START_RANGE client126a socket4031 620
251875-251937
START_RANGE client126a socket4032 630
251937-252000
START_RANGE client127a socket4033 620
252000-252062
START_RANGE client127a socket4034 630
252062-252125
START_RANGE client127a socket4035 620
252125-252187
START_RANGE client127a socket4036 630
252187-252250
START_RANGE client127a socket4037 620
252250-252312
START_RANGE client127a socket4038 630
252312-252375
START_RANGE client127a socket4039 620
252375-252437
START_RANGE client127a socket4040 630
252437-252500
START_RANGE client127a socket4041 620
252500-252562
START_RANGE client127a socket4042 630
252562-252625
START_RANGE client127a socket4043 620
252625-252687
START_RANGE client127a socket4044 630
252687-252750
START_RANGE client127a socket4045 620
252750-252812
START_RANGE client127a socket4046 630
252812-252875
START_RANGE client127a socket4047 620
252875-252937
START_RANGE client127a socket4048 630
252937-253000
START_RANGE client127a socket4049 620
253000-253062
START_RANGE client127a socket4050 630
253062-253125
START_RANGE client127a socket4051 620
253125-253187
START_RANGE client127a socket4052 630
253187-253250
START_RANGE client127a socket4053 620
253250-253312
```

```
START_RANGE client127a socket4054 630
253312-253375
START_RANGE client127a socket4055 620
253375-253437
START_RANGE client127a socket4056 630
253437-253500
START_RANGE client127a socket4057 620
253500-253562
START_RANGE client127a socket4058 630
253562-253625
START_RANGE client127a socket4059 620
253625-253687
START_RANGE client127a socket4060 630
253687-253750
START_RANGE client127a socket4061 620
253750-253812
START_RANGE client127a socket4062 630
253812-253875
START_RANGE client127a socket4063 620
253875-253937
START_RANGE client127a socket4064 630
253937-254000
START_RANGE client128a socket4065 620
254000-254062
START_RANGE client128a socket4066 630
254062-254125
START_RANGE client128a socket4067 620
254125-254187
START_RANGE client128a socket4068 630
254187-254250
START_RANGE client128a socket4069 620
254250-254312
START_RANGE client128a socket4070 630
254312-254375
START_RANGE client128a socket4071 620
254375-254437
START_RANGE client128a socket4072 630
254437-254500
START_RANGE client128a socket4073 620
254500-254562
START_RANGE client128a socket4074 630
254562-254625
START_RANGE client128a socket4075 620
254625-254687
START_RANGE client128a socket4076 630
254687-254750
START_RANGE client128a socket4077 620
254750-254812
START_RANGE client128a socket4078 630
254812-254875
START_RANGE client128a socket4079 620
254875-254937
START_RANGE client128a socket4080 630
254937-255000
START_RANGE client128a socket4081 620
255000-255062
START_RANGE client128a socket4082 630
255062-255125
START_RANGE client128a socket4083 620
255125-255187
START_RANGE client128a socket4084 630
255187-255250
START_RANGE client128a socket4085 620
255250-255312
START_RANGE client128a socket4086 630
255312-255375
START_RANGE client128a socket4087 620
255375-255437
START_RANGE client128a socket4088 630
255437-255500
START_RANGE client128a socket4089 620
255500-255562
START_RANGE client128a socket4090 630
255562-255625
START_RANGE client128a socket4091 620
255625-255687
START_RANGE client128a socket4092 630
255687-255750
START_RANGE client128a socket4093 620
255750-255812
START_RANGE client128a socket4094 630
255812-255875
START_RANGE client128a socket4095 620
255875-255937
START_RANGE client128a socket4096 630
255937-256000
#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
```

RTE Scripts

tpccWeb.h

```
/*
**************************************************
*************
** Project     : AIX DB/2 TPC-C
** Component   : TPC-C/Client
** Name        : tpccWeb.h
** Title       : rte web defines
**************************************************
*************
** Copyright (c) IBM US - AUSTIN 2000
** Classification : IBM Internal Use Only
**
** History       :
**              Develop by Austin RISC/6000
Performance Team
**
** Comments      :
**
**************************************************
*************
*/
//////////////////////////////////////////////////////////
// Transaction Codes
//////////////////////////////////////////////////////////
#define        TXN_LOGIN
#define        TXN_NEW_ORDER
#define        TXN_PAYMENT
#define        TXN_ORDER_STATUS
#define        TXN_DELIVERY
#define        TXN_STOCK
#define        TXN_EXIT                          6
#define        TXN_LOGIN_RESULTS
#define
TXN_NEW_ORDER_RESULTS            8
#define        TXN_PAYMENT_RESULTS
#define
TXN_ORDER_STATUS_RESULTS       10
#define        TXN_DELIVERY_RESULTS        11
#define        TXN_STOCK_RESULTS           12
#define        TXN_NORD                 "nord"
#define        TXN_PYMT                 "pymt"
#define        TXN_ORDS                 "ords"
#define        TXN_DLVY                 "dlvy"
#define        TXN_STOK                 "stok"
#define        TXN_sEXIT                "exit"
#define        TXN_MENU                 "menu"
#define
ITEM_CMD_ID_START                11
#define
ITEM_CMD_ID_END                  55
#define        APP_NAME              "tpcc"
//////////////////////////////////////////////////////////
// Transaction Result Search Strings
//////////////////////////////////////////////////////////
#define            LOGIN_TITLE              "Home Pag
#define            MENU_TITLE               "Main Menu
#define        NORD_TITLE
        "New Order"
#define            PYMT_TITLE               "Payment"
#define            ORDS_TITLE               "Order Statu
#define            DLVY_TITLE               "Delivery"
#define            STOK_TITLE               "Stock Leve
#define            LOGIN_TITLE_LEN          9
#define            MENU_TITLE_LEN               9
#define        NORD_TITLE_LEN               9
#define            PYMT_TITLE_LEN               7
#define            ORDS_TITLE_LEN               12
#define            DLVY_TITLE_LEN               8
#define            STOK_TITLE_LEN               10
#define            NORD_RESULTS_TITLE       "New Order
#define            PYMT_RESULTS_TITLE       "Payment R
#define            ORDS_RESULTS_TITLE       "Order Statu
#define            DLVY_RESULTS_TITLE       "Delivery Re
#define            STOK_RESULTS_TITLE       "Stock Leve
#define
NORD_RESULTS_TITLE_LEN                  17
#define
        PYMT_RESULTS_TITLE_LEN               15
#define
        ORDS_RESULTS_TITLE_LEN               20
#define
        DLVY_RESULTS_TITLE_LEN               16
#define
        STOK_RESULTS_TITLE_LEN               19
#define            CONTENT_LENGTH           "Content-Le
#define            HEADER_TERMINATOR        "\r\n\r\n"

//////////////////////////////////////////////////////////
// Field Lengths
//////////////////////////////////////////////////////////
#define
        HEADER_TERMINATOR_LENGTH             4
#define
        CONTENT_LENGTH_STR_LEN               16
//////////////////////////////////////////////////////////
// Transaction Request URLs
//////////////////////////////////////////////////////////
#define      GET_REQUEST                    "GET %s H
#define      GET_REQUEST_EXIT
        "GET %s HTTP/1.1\r\nHost:
%s\r\nConnection: close\r\nAccept:
text/*\r\n\r\n"
#define      LOGIN_URL                      "/tpcc/tpcc.h
#define      MENU_URL                           "/tpcc/
#define      NEW_ORDER_FORM_URL             "/tpcc/tpcc.h
#define      PAYMENT_FORM_URL                   "/tpcc/
0 #define      ORDER_STATUS_FORM_URL          "/tpcc/tpcc.h
1 #define      DELIVERY_FORM_URL                  "/tpcc/
2 #define      STOCK_FORM_URL                     "/tpcc/
#define      EXIT_FORM_URL                      "/tpcc/
3
4 #define      NEW_ORDER_RESULTS_URL          "/tpcc/tpcc.h
5 #define
        PAYMENT_RESULTS_CID_URL
7       "/tpcc/tpcc.html?00=pymt&01=%d&03
=%d&04=%d&05=&06=%d&07=%d&08=%d
.%02.2d"
9 #define
        PAYMENT_RESULTS_CLAST_URL          "/tpcc/tpcc.h
#define
        ORDER_STATUS_RESULTS_CLAST
_URL
"/tpcc/tpcc.html?00=ords&01=%d&03=%d&
04=&05=%s"
#define
        ORDER_STATUS_RESULTS_CID_U
RL
"/tpcc/tpcc.html?00=ords&01=%d&03=%d&
04=%d&05="
#define  DELIVERY_RESULTS_URL
"/tpcc/tpcc.html?00=dlvy&01=%d&10=%d"
```

```
#define  STOCK_RESULTS_URL
"/tpcc/tpcc.html?00=stok&01=%d&09=%d"
#define      NEW_ORDER_ITEM              "&%d=%d"
#define      NEW_ORDER_ITEM_ENTRY        "&%d=%d&%d=%d"
#define      NEW_ORDER_EMPTY_ITEM        "&%d="
```

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 <strings.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++) {
        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()/100
0.0,

        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)*1
0); 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 */
    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_SIZ
E", &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;
}
*/
```

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;
extern char *expect_buffer_return();
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 nu
const char *CUSTOMER_ID_STRING    =        "Customer
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 *results_request;
    char *form_request;
} gen_tran_t;
typedef struct gen_tran_url_s
{
        char *txn_form_url;
        char *txn_results_url;
}gen_tran_url_s;
int generic_transaction( gen_tran_t
*data,char *host)
{
    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);
          //send menu request page
//iprint(IPRINT_TRACE, "> transmit data-
>menu: (%s)\n request :(%s)",data-
>menu,data->form_request);
//iprint(IPRINT_TRACE, "> transmit data-
>menu: (%s)\n request :(%s)",data->menu);
    transmit(data->form_request);

echo_trace ("Waiting for Menu");
          switch (*data->menu)
          {
                case '1':
                        rc =
expect_html(NORD_TITLE,timeout,NORD_
TITLE_LEN);
                        break;
                case '2':
                        rc =
expect_html(PYMT_TITLE,timeout,PYMT_T
ITLE_LEN);
                        break;
            case '3':
                        rc =
expect_html(ORDS_TITLE,timeout,ORDS_
TITLE_LEN);
                        break;
                case '4':
                        rc =
expect_html(DLVY_TITLE,timeout,DLVY_TI
TLE_LEN);
                        break;
                case '5':
                        rc =
expect_html(STOK_TITLE,timeout,STOK_T
ITLE_LEN);
                        break;
                default:
                        rc = ERROR;
          }
          if(rc == ERROR)
          {
                iprint (IPRINT_ERROR, "Slave
%d:Failed to receive %s input
form\n**Request:-->%s<--\n",
                        shmentry->num, data-
>menu,data->form_request);
                return (ERROR);
          }
//if (expect_html(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
request :(%s)\n",data->results_request);
    transmit(data->results_request);
          echo_trace ("Wait for Response");
          switch (*data->menu)
          {
                case '1':
                        rc =
expect_html(NORD_RESULTS_TITLE,time
out,NORD_RESULTS_TITLE_LEN);
                        break;
                case '2':
                        rc =
expect_html(PYMT_RESULTS_TITLE,timeo
ut,PYMT_RESULTS_TITLE_LEN);
                        break;
                case '3':
                        rc =
expect_html(ORDS_RESULTS_TITLE,time
out,ORDS_RESULTS_TITLE_LEN);
                        break;
                case '4':
                        rc =
expect_html(DLVY_RESULTS_TITLE,timeo
ut,DLVY_RESULTS_TITLE_LEN);
                        break;
                case '5':
                        rc =
expect_html(STOK_RESULTS_TITLE,timeo
ut,STOK_RESULTS_TITLE_LEN);
                        break;
                default:
                        rc = ERROR;
          }
          if(rc == ERROR)
          {
                iprint (IPRINT_ERROR, "Slave
%d:Failed to receive %s result
page\n**Request:-->%s<--\n",
                        shmentry->num, data-
>menu,data->form_request);
                return (ERROR);
          }

// if (expect_html(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_buffer_search("ERROR:",6))
          {
            FILE *fd;
            if ((fd =
fopen("/u/rte/error.txt","a"))!=NULL)
              {
              fprintf(fd,"Slave %d: %s found
'%s'\n%s\n",
                shmentry->num, data->menu,
"ERROR:",expect_buffer_return());
                fclose(fd);
              }
          data->invalid = 1;
          iprint (IPRINT_ERROR, "Slave %d:
%s found '%s'\n%s\n",
                shmentry->num, data->menu,
"ERROR:",expect_buffer_return());
          // 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(shmgloba
l->think[data->type])*1000.0);
    //iprint(IPRINT_TRACE, "<
generic_transaction finish\n");
    return (RTE_OK);
}
/*****************************************************
*************************/
/***                     Delivery Transaction
***/
/*****************************************************
*************************/
int
Delivery(char *host,int terminal)
{
    static struct delivery_struct delivery,
delivery_new;
    int             rc;
    char            *ptr;
    gen_tran_t      tran;
    tran.invalid = 0;
    tran.data    = &delivery;
    tran.len     = sizeof(delivery);
    tran.keywait = 2;
    tran.type    = DELIVERY;
    tran.menu    = "4";
        char  dlvy_url[128];
```

```
        char  form_buffer[256];
        char  results_buffer[256];
        tran.form_request = form_buffer;
        tran.results_request = results_buffer;
        //create dlvy form request
        sprintf(dlvy_url,DELIVERY_FORM_U
RL,terminal);
        sprintf(form_buffer,GET_REQUEST,dl
vy_url,host);
    // Set up all data for new transactions
   delivery_new.carrier = uniform(1, 10);  //
carrier # 1 to 10
        //create dlvy results request
        sprintf(dlvy_url,DELIVERY_RESULTS
_URL,terminal,delivery_new.carrier);
        sprintf(results_buffer,GET_REQUEST,
dlvy_url,host);

    // Go do the transaction
    rc = generic_transaction(&tran,host);
    delivery         = delivery_new;
    delivery.invalid = tran.invalid;
         //iprint(IPRINT_TRACE,"dlvy txn
finished, rc:%d
tran.invalid:%d\n",rc,delivery.invalid);
    return (rc);
}
/*************************************************
*************************/
/***                New Order Transaction
***/
/*************************************************
*************************/
int NewOrder(char *host,int terminal)
{
   static struct neword_struct neword,
neword_new;
   int            i, rc, whses, low_whse=1;
        char  nord_form_url[128];
        char  form_buffer[512];
        char  nord_results_url[2048];
        char  results_buffer[4096];
   char          *ptr;
   char                *ptr2;
        const char          *err_ptr;
   gen_tran_t      tran;
   tran.invalid                = 0;
   tran.data                   = &neword;
   tran.len                    = sizeof(neword);
   tran.keywait                = 18;
   tran.type                   = NEWORDER;
   tran.menu                   = "1";

   tran.form_request      = form_buffer;
        tran.results_request     =
results_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);
   neword_new.olremote       =       0;
   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);
   //create new order form request
        sprintf(nord_form_url,NEW_ORDER_
FORM_URL,terminal);

        //create get form request
        sprintf(form_buffer,GET_REQUEST,n
ord_form_url,host);
        //create new order results url
        char itemString[1024];
        ptr2=itemString;
        short item_cmd_start =
ITEM_CMD_ID_START;
        for (i = 0; i < neword_new.nloop; i++)
        {
                ptr2 += sprintf(ptr2,
NEW_ORDER_ITEM,

item_cmd_start++,
                        neword_new.item[i].olswid);

            ptr2 +=
sprintf(ptr2,NEW_ORDER_ITEM,

item_cmd_start++,
                        neword_new.item[i].oliid);

                ptr2 += sprintf(ptr2,
NEW_ORDER_ITEM,

item_cmd_start++,

neword_new.item[i].olquantity);
        }
        //seal up url w/ empty items
        for (i = item_cmd_start;i <=
ITEM_CMD_ID_END; i++)
        {
                ptr2 +=
sprintf(ptr2,NEW_ORDER_EMPTY_ITEM,i);
        }

        sprintf(nord_results_url,NEW_ORDER
// number of ITEM URL generated
        RESULTS_URL,terminal);
                                // find total number of remote order-lines
neword_new.did,neword_new.cid,

itemString);

        //create get results request
        sprintf(results_buffer,GET_REQUEST,
nord_results_url,host);

   // Go do the transaction
   rc = generic_transaction(&tran,host);
   neword         = neword_new;
   neword.invalid = tran.invalid;
   // Check for a rollback
   if ((err_ptr = expect_buffer_search("Item
number is not valid",24)))
        {
                neword.rollback=1;
                echo_trace ("Found rollback!\n");
   }
   // Grab the orderID from the
   if (!(err_ptr = expect_buffer_search("Order
Number: ",14)))
        {
                echo_trace ("Didn't find order-id
for neworder");
                iprint (IPRINT_ERROR,
"Neworder didn't have Order-
ID\n%s\n",expect_buffer_return());
                //iprint (IPRINT_ERROR,
"Neworder didn't have Order-ID\n");
                neword.oid = -1;
   }
        else
        {
                neword.oid = atoi(err_ptr+14);
                //
        iprint(IPRINT_ERROR,"New order
order id:%d\n",neword.oid);
   }


   // 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_T
IMES,
keystroke_length*sizeof(int),keystroke_time
s);
   }
         //iprint(IPRINT_TRACE,"nord txn
finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);
   return (rc);
}

/*************************************************
*************************/
/***                Order Status Transaction
***/
/*************************************************
*************************/
int OrderStatus(char *host,int terminal) {
   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;

        //Joe N.
        char  ords_url[256];
        char  form_buffer[512];
        char  results_buffer[2048];
        tran.results_request     =
results_buffer;
   tran.form_request = form_buffer;

        //create order status form request
        sprintf(ords_url,ORDER_STATUS_FO
RM_URL,terminal);
        sprintf(form_buffer,GET_REQUEST,or
ds_url,host);

   // 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;
  }
        //iprint(IPRINT_TRACE,"Order status
fields,w_id:%d d_id:%d  n_d_id:%d c_id:%d
\n", data->Warehouse,data-
>District,ordstat_new.did,ordstat_new.cid);
        //create order status url request
        if (ordstat_new.byname)

        sprintf(ords_url,ORDER_STATUS_RE
SULTS_CLAST_URL,terminal,ordstat_new.
did,

        else

        sprintf(ords_url,ORDER_STATUS_RE
SULTS_CID_URL,terminal,ordstat_new.did,

        sprintf(results_buffer,GET_REQUEST,
ords_url,host);

    // Go do the transaction
    rc = generic_transaction(&tran,host);
    ordstat         = ordstat_new;
    ordstat.invalid = tran.invalid;
         //iprint(IPRINT_TRACE,"ords txn
finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);
    return (rc);
}
/*****************************************************
************************/
/***                    Payment Transaction
***/
/*****************************************************
************************/
int
Payment(char *host,int terminal)
{
    static struct payment_struct payment,
payment_new;
    int             dollars, cents, rc, whses,
low_whse = 1;
    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";

        char pymt_url[128];
        char form_buffer[256];
        char results_buffer[2048];
        tran.results_request     =
results_buffer;
   tran.form_request = form_buffer;
        //create pymt form url
        sprintf(pymt_url,PAYMENT_FORM_U
RL,terminal);
        sprintf(form_buffer,GET_REQUEST,p
ymt_url,host);

    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 */             ordstat_new.clast);
        payment_new.cwid = (long)
uniform((long)low_whse, (long) whses-1);
        if (payment_new.cwid >= WHSEID)
            payment_new.cwid++;
        payment_new.remote = 1;          ordstat_new.cid);
        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;

        //create payment results url
        if (payment_new.byname)

        sprintf(pymt_url,PAYMENT_RESULT
S_CLAST_URL,terminal,

payment_new.did,payment_new.clast,paym
ent_new.cwid,

payment_new.cdid,dollars,cents);
        else

        sprintf(pymt_url,PAYMENT_RESULT
S_CID_URL,terminal,

payment_new.did,payment_new.cid,payme
nt_new.cwid,

payment_new.cdid,dollars,cents);
        sprintf(results_buffer,GET_REQUEST,
pymt_url,host);

    // Go do the transaction
    rc = generic_transaction(&tran,host);
    payment         = payment_new;
    payment.invalid = tran.invalid;
         //iprint(IPRINT_TRACE,"pymt txn
finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);

        return (rc);
}
/*****************************************************
************************/
/***                    Stock Level Transaction
***/
/*****************************************************
************************/
int
StockLevel(char *host,int terminal)
{
    static struct stocklev_struct stocklevel,
stocklevel_new;
    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";
        char stok_url[128];
        char form_buffer[256];
        char results_buffer[2048];
        tran.results_request     =
results_buffer;
   tran.form_request = form_buffer;
        //create stok form url
        sprintf(stok_url,STOCK_FORM_URL,t
erminal);
        sprintf(form_buffer,GET_REQUEST,st
ok_url,host);

    stocklevel_new.invalid = 0;
    stocklevel_new.threshold = uniform(10,
20);   /* uniform no. between 10 and
                                        * 20 */
        //create stok results url
        sprintf(stok_url,STOCK_RESULTS_U
RL,terminal,stocklevel_new.threshold);
        sprintf(results_buffer,GET_REQUEST,
stok_url,host);

    // Go do the transaction
    rc = generic_transaction(&tran,host);
    stocklevel         = stocklevel_new;
    stocklevel.invalid = tran.invalid;
         //iprint(IPRINT_TRACE,"stok txn
finished, rc:%d
tran.invalid:%d\n",rc,tran.invalid);
    return (rc);
}
/*****************************************************
************************/
/***  MAIN()
***/
/*****************************************************
************************/
int
user_transaction(char *host,void *data,int
terminal)
{
        UserLocal *localdata = (UserLocal
*)data;
    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(host,terminal);
```

```
                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(host,terminal);
                        case 1: return
OrderStatus(host,terminal);
                        case 2: return
Payment(host,terminal);
                        case 3: return
StockLevel(host,terminal);
                        case 4: task = 0; return
NewOrder(host,terminal);
                        return
NewOrder(host,terminal);
                }
  }

/*****************************************************
*************************/
  /***  CHOOSE ONE OF THE
TRANSACTIONS                                  ***/

/*****************************************************
*************************/
  ntask = (double) uniform(0.0, 100.0);
  if (ntask <= shmglobal-
>chances[DELIVERY])
        {
        return Delivery(host,terminal);
                //return
NewOrder(host,terminal);
        }
  ntask -= shmglobal->chances[DELIVERY];
  if (ntask <= shmglobal-
>chances[ORDSTAT])
  {
                return
OrderStatus(host,terminal);
  }
  ntask -= shmglobal->chances[ORDSTAT];
  if (ntask <= shmglobal-
>chances[PAYMENT])
        {
                return Payment(host,terminal);
  }
  ntask -= shmglobal->chances[PAYMENT];
  if (ntask <= shmglobal-
>chances[STOCKLEV])
        {
                return StockLevel(host,terminal);
  }
  return NewOrder(host,terminal);
#else
   {
        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(host,terminal);
        case OrderStatus: return
OrderStatus(host,terminal);
        case Payment:     return
Payment(host,terminal);
        case StockLevel:  return
StockLevel(host,terminal);
        case NewOrder:    return
NewOrder(host,terminal);
        }
        /*
        switch(tmp) {
        case Delivery:    return
Payment(host,terminal);
        case OrderStatus: return
Payment(host,terminal);
        case Payment:     return
Payment(host,terminal);
        case StockLevel:  return
Payment(host,terminal);
        case NewOrder:    return
NewOrder(host,terminal);
        */
        }
   }
#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_login_html(char *host,void *data,int
*terminal)
{
   UserLocal *localdata = (UserLocal *)data;
   int rc;
   int timeout_value = shmglobal-
>login_timeout;
   char request[256];
        char url[30];
   set_typing_delay(0);
iprint(IPRINT_ERROR,"Generating login
request for host:%s\n",host);
   //generate login page request

sprintf(request,GET_REQUEST,LOGIN_UR
L,host);
iprint(IPRINT_ERROR,"sending login form
request:%s\n",request);
        //send the request
        transmit(request);
iprint(IPRINT_ERROR,"login request sent,
reading response in expect_html()\n");
        //read the request
        rc =
expect_html(LOGIN_TITLE,timeout_value,L
OGIN_TITLE_LEN);
        if (rc != RTE_OK)
        {
                iprint(IPRINT_ERROR,"Login
request failed, unable to find login key
words:%s\n",LOGIN_TITLE);
                return RTE_ERROR;
        }

iprint(IPRINT_ERROR,"login request
read\n");
        //generate url and page get request
        sprintf(url,MENU_URL,localdata-
>Warehouse, localdata->District);

        sprintf(request,GET_REQUEST,url,ho
st);
iprint(IPRINT_ERROR,"sending login results
request:%s\n",request);
   transmit(request);
iprint (IPRINT_TRACE, "Slave %d:
Warehouse=%d, District=%d, pid=%d\n",
shmentry->num, localdata->Warehouse,
localdata->District, getpid());
   rc = expect_html(MENU_TITLE,
timeout_value,MENU_TITLE_LEN);
   if (rc != RTE_OK)
        {
       iprint (IPRINT_ERROR, "Slave %d:
Failed logging in\n", shmentry->num);
       return RTE_ERROR;
   }
iprint (IPRINT_TRACE, "User login
successful Slave%d: Warehouse=%d,
```

```
District=%d, pid=%d\n",shmentry->num,
localdata->Warehouse, localdata->District,
getpid());
        rc = get_term_id(terminal);
        if(rc != RTE_OK)
                return RTE_ERROR;
iprint(IPRINT_TRACE, "Terimnal set for this
user:%d w/ warehouse:%d
district:%d\n",*terminal,localdata-
>Warehouse,localdata->District);

   return RTE_OK;
}
int get_term_id(int *terminal)
{
        //search for terminal id
        const char *termID_ptr;
   if (!(termID_ptr =
expect_buffer_search("NAME=\"01\"
VALUE=\"",17)))
        {
                echo_trace ("Did not find
terminal id in response....");
                iprint (IPRINT_ERROR, "No
terminal id specified.");
                return RTE_ERROR;
        }
        else
        {
                *terminal = atoi(termID_ptr+17);
                iprint(IPRINT_ERROR,"Terminal
id:%d\n",terminal);
        }
        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 () {

    iprint (IPRINT_TRACE, "Slave %d:
Warehouse=%d, District=%d logging out\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;
}
```

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
/*****************************************
*****************************************/
/***   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 */
};
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;
};
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 UserGlobal *shmglobal;
extern UserLocal  *shmlocal;
#endif
```

Appendix - E: Pricing

Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

Tel 425 882 8080
Fax 425 936 7329
http://www.microsoft.com/

Microsoft

August 2, 2005

IBM Corporation
Tony Petrossian
11501 BURNET ROAD
Austin, TX 78758

Mr. Petrossian:

Here is the information you requested regarding pricing for several Microsoft products to be used in conjunction with your TPC-C benchmark testing.

All pricing shown is in US Dollars ($).

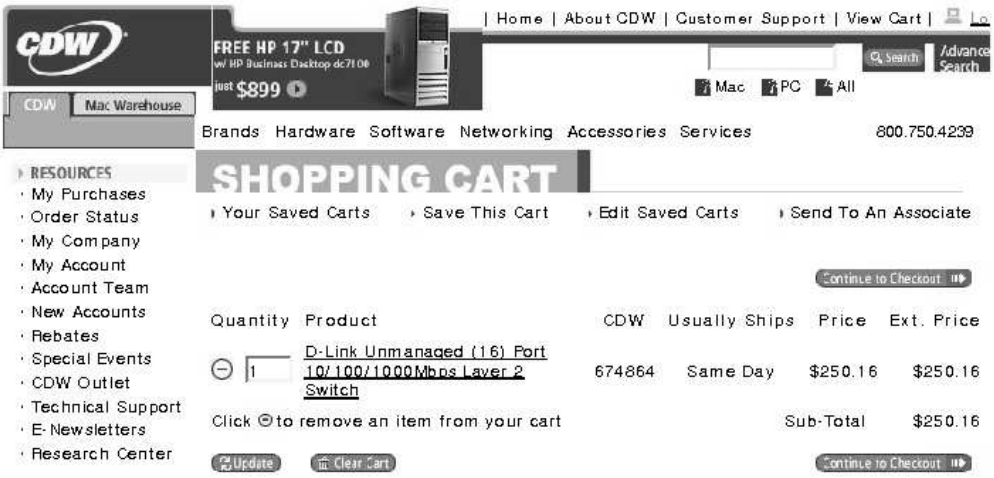
Part Number	Description	Unit Price	Quantity	Price
C11-00821	**Windows 2000 Server** *Server License Only - No CALs* *Discount Schedule: No Level* *Unit Price reflects a 8% discount from the retail unit price of $799.*	$738	8	$5,904
254-00170	**Visual C++ Standard Edition** *Discount Schedule: No Discounts Applied*	$109	1	$109
	Microsoft Problem Resolution Services *Professional Support* *(1 incident)*	$245	1	$245

All products are currently orderable through Microsoft's normal distribution channels.

This quote is valid for the next 90 days.

If we can be of any further assistance, please contact Jamie Reding at (425) 703-0510 or jamiere@microsoft.com.

Reference ID: PCtope0502088897.
Please include this Reference ID in any correspondence regarding this price quote.


| Home | About CDW | Customer Support | View Cart |
FREE HP 17" LCD
w/ HP Business Desktop dc7100
just $899
Search
Advance Search
Mac PC All
CDW
Mac Warehouse
Brands Hardware Software Networking Accessories Services
800.750.4239
RESOURCES
My Purchases
Order Status
My Company
My Account
Account Team
New Accounts
Rebates
Special Events
CDW Outlet
Technical Support
E-Newsletters
Research Center
SHOPPING CART
Your Saved Carts
Save This Cart
Edit Saved Carts
Send To An Associate
Continue to Checkout
Quantity Product CDW Usually Ships Price Ext. Price
1 D-Link Unmanaged (16) Port 10/100/1000Mbps Layer 2 Switch 674864 Same Day $250.16 $250.16
Click to remove an item from your cart
Sub-Total $250.16
Update
Clear Cart
Continue to Checkout

redhat

Red Hat Store
Shopping Cart

'Red Hat Enterprise Linux AS (Standard for x86, AMD64 & Intel EM64T, Intel Itanium, and POWER)' is already in your order

Contracts

	List	Contract price	Change quantity	Remove	Line total
New Contract	Aug. 5, 2005	Aug. 4, 2006			
Red Hat Enterprise Linux AS (Standard for x86, AMD64 & Intel EM64T, Intel Itanium, and POWER)	$1,499.00	$1,499.00	1	☐	$1,499.00

Add to this contract:

Browse store

Subtotal: $1,499.00

International Business Machines Corporation

3200 Windy Hill Road
Atlanta, Georgia 30339

August 8, 2005

Berni Schiefer/Ray Venditti
IBM Corporation

Dear Berni/Ray:

The requested quote for the TPC-C Linux on POWER Benchmark is below.

All pricing shown is in US Dollars ($)

Product	Description	Qty	List Price	Extended Purchase	3yr Maint 24x7
Server Hardware					
9117-570	pSeries p5 9117 Model 570	1	$2,300.00	$2,300.00	$3,072.00
6942-25B	9117-570 24x7x4 WSU	1	$4,070.00	$0.00	$4,074.00
1846	Op Panel	1	$150.00	$150.00	
2640	IDE Slimline DVD-ROM Drive	1	$378.00	$378.00	
3274	73.4 GB 10,000 RPM Ultra320 SCSI Disk Drive Assembly	2	$750.00	$1,500.00	
4492	32768MB (4x8192MB) DIMMs, 208-pin, 200MHz Stacked DDR1 SDRAM	4	$103,000.00	$412,000.00	
4651	Rack Indicator, Rack #1	1	N/C	N/C	
5716	2 Gigabit Fibre Channel PCI-X Adapter	5	$2,267.00	$11,335.00	
6458	Power Cable -- Drawer to IBM PDU, 14-foot, 250V/10A	2	$14.00	$28.00	
7164	IBM Rack-mount Drawer Rail Kit	1	$168.00	$168.00	
7832	Processor, 0/2 Way 1.9 GHz with DDR1 memory	2	$10,900.00	$21,800.00	$7,680.00
7865	CEC Backplane	1	$1,200.00	$1,200.00	
7866	I/O Backplane	1	$4,100.00	$4,100.00	
7867	Midplane	1	$500.00	$500.00	
7868	DASD Backplane	1	$1,200.00	$1,200.00	
7869	Media Backplane	1	$140.00	$140.00	
7870	Power Midplane	1	$200.00	$200.00	
7875	CPU Regulator	3	$1,100.00	$3,300.00	
7878	Serial Port Riser Card	1	$100.00	$100.00	
7879	System Drawer Enclosure	1	$350.00	$350.00	
7881	Service Processor Card (FSP)	1	$650.00	$650.00	
7888	AC Power Supply, 240V, 1400 W	2	$800.00	$1,600.00	
7898	Single Processor Activation For #7832	2	$21,800.00	$43,600.00	$6,768.00
8454	Zero-priced Processor Activation Code for #7832	2		N/C	$6,768.00
9300	Language Group Specify - US English	1		N/C	
			Subtotal	**$506,599.00**	**$28,362.00**
7014-T00	Rack 1:Rack Model T00	1	$2,920.00	$2,920.00	$768.00
0231	Rack Content Specify: 9117/570 - 4EIA	1	N/C	N/C	
4651	Rack Indicator, Rack #1	1	N/C	N/C	
6246	1.8m Rack Trim Kit	1	$158.00	$158.00	
6654	PDU to Wall Powercord 14', 200-240V/24A, UTG0247, PT#12	1	$240.00	$240.00	
9188	Power Distrib Unit Specify - Base/Side Mount, Universal UTG0247 Connector	1	N/C	N/C	
9300	Language Group Specify - US English	1	N/C	N/C	

			Subtotal	**$3,318.00**	**$768.00**
7310-C04	HMC 1:7310-C04 Desktop Hardw.Mgmt.Console	1	$1,830.00	$1,830.00	$1,152.00
6942-25B	7310-C04 24x7x4 WSU	1	$192.00		$192.00
0961	Hardware Management Console for POWER5 Licensed Machine Code	1	N/C	N/C	
3637	IBM T541H /L150p 15 inch TFT Color Monitor	1	$625.00	$625.00	
6470	Power Cord (6-foot), To Wall (125V, 15A), Plug Type #4	2	$14.00	$28.00	
7801	Ethernet Cable, 6M, Hardware Management Console to System Unit	1	$12.00	$12.00	
8800	Quiet Touch Keyboard - USB, Business Black, US English, #103P	1	$83.00	$83.00	
8841	Mouse - Business Black with Keyboard Attachment Cable	1	$62.00	$62.00	
9300	Language Group Specify - US English	1	N/C	N/C	
			Subtotal	**$2,640.00**	**$1,344.00**
Storage					
1722-60U	DS4300 Midrange Disk (60U)	1	$14,999.00	$14,999.00	
6942-25B	1722-60U 24x7x4 WSU	1	$2,499.00		$2,499.00
2210	(19K1271) Short Wave SFP GBIC (LC)	2	$499.00	$998.00	
5213	(32P0768) 2Gb FC, 73.4GB/15K Drive Module	5	$2,099.00	$10,495.00	
5605	(19K1248) 5M Fibre Optic Cable (LC-LC)	1	$129.00	$129.00	
7714	(25R0091) DS4300 Linux for Power Host Kit	1	$4,000.00	$4,000.00	
9202	Field Integrate DS4300	1	N/C	N/C	
9800	(6952300) Power Cord 125V/10A, (Group 1)	1	N/C	N/C	
			Subtotal	**$30,621.00**	**$2,499.00**
1740-710	DS4000 EXP710 Storage Expansion	24	$6,000.00	$144,000.00	
6942-25B	1740-710 24x7x4 WSU	24	$760.00		$18,240.00
2210	(19K1271) 2Gb Fibre Channel Short Wave GBIC	92	$499.00	$45,908.00	
5212	(06P5772) 2Gb FC, 36.4GB/15K Drive	320	$1,115.00	$356,800.00	
5601	(19K1247) 1m 50u Fiber Optic Cable (LC-LC)	48	$79.00	$3,792.00	
9009	Attach to the DS4500 (1742-90U)	24	N/C	N/C	
9202	Field Integrate FC EXP 710	8	N/C	N/C	
9800	(6952300) Power Cord 125V/10A, (Group 1)	24	N/C	N/C	
			Subtotal	**$550,500.00**	**$18,240.00**
1742-90U	DS4500 Midrange Disk System	2	$49,900.00	$99,800.00	
6942-25B	1742-90U 24x7x4 WSU	2	$1,087.00		$2,174.00
2210	(19K1271) 2Gb Fibre Channel Short Wave GBIC	8	$499.00	$3,992.00	
3507	(19K1269) DS4000 2 Gbps miniHUB (LC)	4	$899.00	$3,596.00	
5605	(19K1248) Fiber Cable 5m Multimode (LC-LC)	8	$129.00	$1,032.00	
9202	Field Integrate DS4500	2	N/C	N/C	
9800	(6952300) Power Cord 125V/10A, (Group 1)	2	N/C	N/C	
			Subtotal	**$108,420.00**	**$2,174.00**
Software					
5773-RS3	Initial Hardw.Mgmt.Console Software Support 3 Year	1	N/A	N/A	
0569	Per Processor Software Support 3 Year	1	$675.00	N/A	$675.00
0570	Per Processor 24x7 Software Support 3 Year	1	$236.00	N/A	$236.00
7000	Agreement for MCRSA	1	N/A	N/A	
	DB2 UDB ESE 8.2 SW Lic & Mtce 12 MO	4	$22,608.00	$90,432.00	
	DB2 UDB ESE SW Mtce Renewal 2 YR	8	$1,077.00		$8,616.00
			Subtotal	**$90,432.00**	**$9,527.00**
Client Hardware and Software					
8837E3U	Express x336, Xeon 2x3.2GHz/800MHz, 2GB, DVD-ROM	8	$4,035.00	$32,280.00	
21P2073	Warranty Service Upgrade: 3 Yr OnSite 24x7x4 hours	8	$450.00		$3,600.00
30R5096	Express IBM 36.4GB 15K U320 SCSI HS Option	8	$275.00	$2,200.00	
93074SX	IBM S2 42U Standard Rack Cabinet	1	$1,489.00	$1,489.00	$300.00
90P0726	IBM C170 17" CRT Monitor	1	$185.00	$185.00	$90.00
25R6968	IBM Preferred Pro Full Size KB - PS/2	1	$29.00	$29.00	
90P0740	IBM Sleek 2-Button Mechanical Ball Mouse - PS/2	1	$15.00	$15.00	
			Subtotal	**$36,198.00**	**$3,990.00**
			Total	**$1,328,728.00**	**$66,904.00**

IBM Total Configuration Discount	$629,430.00
Total 3 Yr Cost of IBM Configuration	$766,202.00

For additional information, please contact me directly at 770-835-6612.

Alycia E. Waller
Technical Sales Specialist - eServer pSeries
awaller@us.ibm.com