TPC Benchmark™C

Full Disclosure Report

Fujitsu
GP7000F Model 600 c/s w/ 6 Front-Ends

running

SymfoWARE Server
Enterprise Edition for VLM 2.0

June 23, 1999

The benchmark results contained in this document were submitted for compliance with version 3.4 of the TPC Benchmark C Standard Specification. The result of that action is to place these benchmark results into the sixty day "under review" status as of June 23, 1999.

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

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

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

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

Printed in the United States June 23, 1999

FUJITSU

Preface

The TPC Benchmark C was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry. This full disclosure report is based on the TPC Benchmark C Standard Specifications Version 3.4, released August 25th, 1998.

TPC Benchmark C Overview

The TPC describes this benchmark in Clause 0.1 of the specifications as follows:

TPC Benchmark C is an On Line Transaction Processing (OLTP) workload. It is a mixture of read-only and update intensive transactions that simulate the activities found in complex OLTP application environments. It does so by exercising a breadth of system components associated with such environments, which are characterized by:

- The simultaneous execution of multiple transaction types that span a breadth of complexity
- On-line and deferred transaction execution modes
- Multiple on-line terminal sessions
- Moderate system and application execution time
- Significant disk input/output
- Transaction integrity (ACID properties)
- Non-uniform distribution of data access through primary and secondary keys
- Databases consisting of many tables with a wide variety of sizes, attributes, and relationships
- Contention of data access and update

The performance metric reported by TPC-C is a “business throughput” measuring the number of orders processed per minute. Multiple transactions are used to simulate the business activity of processing an order, and each transaction is subject to a response time constraint. The performance metric for this benchmark is expressed in transactions-per-minute-C (tpmC). To be compliant with the TPC-C standard, all references to tpmC results must include the tpmC rate, the associated price-per-tpmC, and the availability date of the priced configuration.

Despite the fact that this benchmark offers a rich environment that emulates many OLTP applications, this benchmark does not reflect the entire range of OLTP requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to other environments are not recommended.

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

FUJITSU

Abstract

Overview

This report documents the methodology and results of the TPC Benchmark C test conducted by Fujitsu Ltd. on the Fujitsu GP7000F Model 600 c/s w/ 6 Front-Ends. The operating system used for the benchmark was Solaris 2.6. The DBMS used was SymfoWARE Server Enterprise Edition for VLM 2.0.

TPC Benchmark C Metrics

The standard TPC Benchmark C metrics, tpmC (transactions per minute), price per tpmC (five year capital cost per measured tpmC), and the availability date are reported as:

33,113.60 tpmC
$83.74 per tpmC
December 23, 1999

Standard and Executive Summary Statements

The following pages contain the executive summary of results for this benchmark.

Auditor

The benchmark configuration, environment and methodology, along with the pricing model used to calculate the cost per tpmC, were audited by Francois Raab of InfoSizing to verify compliance with the relevant TPC specifications.

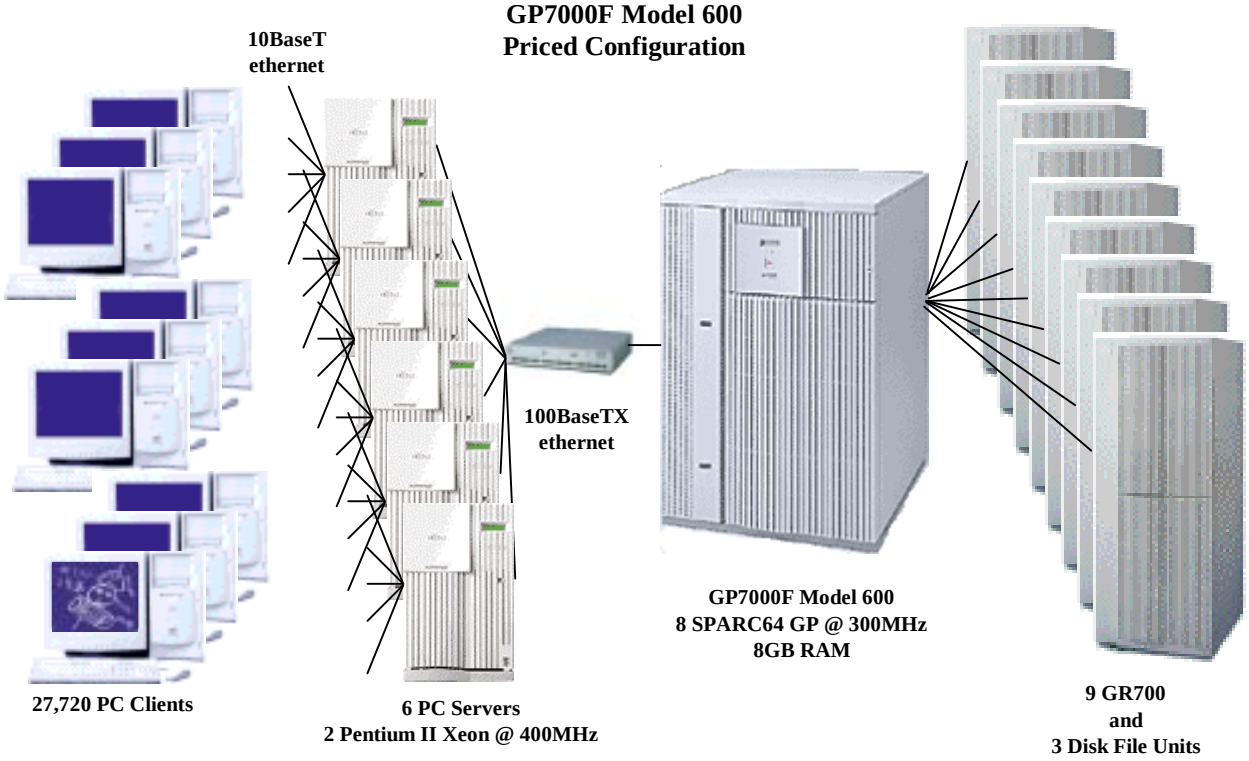
Priced Configuration

FUJITSU	GP7000F Model 600 c/s w/ 6 Front-Ends	TPC-C Rev 3.4 Report Date: June 23, 1999

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$2,772,999	33,113.60 tpmC	$83.74/tpmC	December 23, 1999

Processors	Database Manager	Operating system	Other Software	Number of users
8 SPARC64 GP @ 300MHz	SymfoWARE Server Enterprise Edition for VLM 2.0	Solaris 2.6	BEA Tuxedo 6.4 CFS, Fujitsu COBOL V4, Microsoft Internet Information Server	27,720

GP7000F Model 600
Priced Configuration



RDBMS SERVER			CLIENTS(EACH OF 6)	
	QTY	DESCRIPTION	QTY	DESCRIPTION
PROCESSOR	8	SPARC64 GP @ 300MHz	2	PENTIUM II XEON @ 400MHz
CACHE MEMORY		8MB (EACH PROCESSOR)		512KB (EACH)
MEMORY		8GB		512MB
DISK CONTROLLER	7	ULTRA WIDE DIFFERENTIAL SCSI (2 CHANNELS)	1	WIDE-SCSI
DISKS	448	9GB (10,000RPM)	1	4GB
	8	4GB (10,000RPM)		
	1	4GB (7,200RPM)		
TOTAL		3826.13GB		
TERMINAL	1	CONSOLE TERMINAL	1	17" MONITOR
NETWORK INTERFACE	1	FASTETHERNET ADAPTER	7	1 FASTETHERNET ADAPTER, 6 ETHERNET ADAPTERS
HUBS	1	24-PORT (100 BASE-TX)	3,495	COMPEX MICROHUB (8 PORTS)

Detailed Pricing Information

FUJITSU

Detailed Pricing Information

GP7000F Model 600 c/s
w/ 6 Front-Ends

TPC-C Rev 3.4

Report Date:

June 23, 1999

Order Number	Description	Quantity	Third Party	Unit Price	Extended Price	Maintenance rate/unit*		5 Years Maintenance
Server Hardware								
GP7F0C1U	GP7000F Model 600 w/2 x SPARC64 GP 300MHz/8MB cache	1		33,289.10	33,289.10	4,993.37	/5yr	4,993.37
GP7F1C12U	2 x SPARC64 GP 300MHz/8MB cache (2 to 4)	1		22,230.00	22,230.00	3,334.50	/5yr	3,334.50
GP7F1C13U	2 x SPARC64 GP 300MHz/8MB cache (4 to 6)	1		22,230.00	22,230.00	3,334.50	/5yr	3,334.50
GP7F1C14U	2 x SPARC64 GP 300MHz/8MB cache (6 to 8)	1		22,230.00	22,230.00	3,334.50	/5yr	3,334.50
GP7F2M61U	1GB (4 x 256MB) Memory	8		5,200.00	41,600.00	780.00	/5yr	6,240.00
GP7B7PB1U	PCI Slot Extension	1		663.00	663.00	99.45	/5yr	99.45
X6541A-U	Ultra/Wide/Differential SCSI-2 adapter	7		1,224.00	8,568.00	183.60	/5yr	1,285.20
GP7B3D41U	Internal HDD (4GB, 7200rpm)	1		552.00	552.00	82.80	/5yr	82.80
GP7B3D42U	Internal HDD (4GB, 10000rpm)	8		780.00	6,240.00	117.00	/5yr	936.00
GP7B3D92U	Internal HDD (9GB, 10000rpm)	23		1,040.00	23,920.00	156.00	/5yr	3,588.00
GP7B7FU1U	Expansion File Unit	3		2,470.00	7,410.00	370.50	/5yr	1,111.50
GP7B'FU92U	File Unit Expansion	3		585.00	1,755.00	87.75	/5yr	263.25
WY-55	Wyse WY-55 terminal + keyboard + 2 year warranty extension	1	1	573.00	573.00	0.00	/yr	0.00
GP723ET2U	8mm Tape unit	1		884.00	884.00	132.60	/5yr	132.60
F6403A3L	Disk array Rackmount model A3 base unit	9		22,202.00	199,818.00	720.00	/mo	388,800.00
	(includes Disk Drive Modules & cables)							
F6403A12L	SCSI RAID controller pair M128 (128MB cache)	18		16,694.00	300,492.00	2,504.10	/5yr	45,073.80
F6403A23L	10,000rpm 9GB disk drive	425		1,687.00	716,975.00	253.05	/5yr	107,546.25
F6403A41L	19inch Rack (Data Center Cabinet)	9		2,890.00	26,010.00	433.50	/5yr	3,901.50
SU3000INET	APC SU3000INET UPS (3000VA/2250W)	1	2	2,252.00	2,252.00	457.00	/5yr	457.00
DCBL-SCR05	5m VHDCI - 68P SCSI cable	16		86.00	1,376.00	12.90	/5yr	206.40
SCBL-DSC05	5m 68P - 68P SCSI cable	18		86.00	1,548.00	12.90	/5yr	232.20
	Server Storage Subtotals				1,440,615.10			574,952.82
Server Software								
	Solaris 2.6 and Server Supplement Software	1		100.00	100.00	0.00	/yr	0.00
B23G350H0	SymfoWARE Server Enterprise Edition for VLM 2.0	1		240,098.00	240,098.00	48,019.60	/yr	240,098.00
	SynfinityDisk 1.0	1		832.00	832.00	0.00	/yr	0.00
	Server Software Subtotals				241,030.00			240,098.00
Client Hardware								
GP5L820	GRANPOWER 5000 model 580	6		5,666.67	34,000.02	850.00	/5yr	5,100.00
	w/ 1 x Pentium II Xeon 400MHz/512KB, 128MB memory, 0GB disk,							
	1 10/100Base-TX							
GP5-FG10D	Pentium II Xeon (400MHz/512KB)	6		1,347.86	8,087.16	202.18	/5yr	1,213.07
GP5-RM12G	128MB (4 x 32MB) Memory	18		295.48	5,318.64	44.32	/5yr	797.80
GP5-HDH45	Internal HDD (4GB)	6		352.14	2,112.84	52.82	/5yr	316.93
GP5-182	FastEthernet Adapter (10/100Base-TX)	36		72.86	2,622.96	10.93	/5yr	393.44
TMSRVR-OPTION-5929	17" Monitor (Universal)	6		465.48	2,792.88	69.82	/5yr	418.93
	Client Hardware Subtotals				54,934.50			8,240.18
Client Software								
	Microsoft Windows NT Server 4.0, include 5 CALs	6	3	809.00	4,854.00	0.00	/yr	0.00
	Microsoft Visual C++ Professional 6.0	1	3	549.00	549.00	0.00	/yr	0.00
	Microsoft Software Maintenance	1	3	0.00	0.00	2,095.00	/yr	10,475.00
	Fujitsu COBOL Standard Edition	1		750.00	750.00	150.00	/yr	750.00
	BEA Tuxedo 6.4 CFS	6	4	3,000.00	18,000.00	450.00	/yr	13,500.00
	Client Software Subtotals				24,153.00			24,725.00
User Connectivity								
	CISCO Catalyst 2924XL 10/100 Ethernet Switch (24ports) *	3	5	3,295.00	9,885.00	294.00	/5yr	4,410.00
TP1008C	8-Port RJ45 & BNC Ethernet Hub *	3845	6	39.00	149,955.00	0.00	/mo	0.00
	User Connectivity Subtotals				159,840.00			4,410.00
	Totals				1,920,572.60			852,425.99
	5 Year cost							2,772,999
	tpmC							33,113.60
	$ / tpmC							83.74
Third Party :	1 = SSP Data Products							
	2 = American Power Conversion Corp.							
	3 = Microsoft Corporation							
	4 = BEA Systems							
	5 = CISCO SYSTEMS							
	6 = COMPEX INC.							
	* : 10% or minimum of 2 spares are included							

Notes:

Results independently audited by Francois Raab of InfoSizing, Inc.

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

Numerical Quantities Summary

GP7000F Model 600 c/s w/ 6 Front-Ends

SymfoWARE Server Enterprise Edition for VLM 2.0

MQTH, Computed Maximum Qualified Throughput			**33,113.60 tpmC**			
Response Times (in seconds)	**Average**		**90%**		**Max.**	
New-Order	**1.26**		**1.60**		**66.23**	
Payment	**0.83**		**1.48**		**66.33**	
Order-Status	**0.80**		**1.46**		**61.07**	
Delivery (interactive portion)	**0.23**		**0.16**		**49.36**	
Delivery (deferred portion)	**1.09**		**1.86**		**61.80**	
Stock-Level	**0.81**		**1.49**		**61.09**	
Menu	**0.40**		**0.15**		**50.24**	
Transaction Mix, in percent of total transaction						
New-Order					**44.79**	
Payment					**43.08**	
Order-Status					**4.06**	
Delivery					**4.03**	
Stock-Level					**4.04**	
Emulation Delay (in seconds)			**Resp. Time**		**Menu**	
New-Order			.1		.1	
Payment			.1		.1	
Order-Status			.1		.1	
Delivery (interactive)			.1		.1	
Stock-Level			.1		.1	
Keying/Think Times (in seconds)	**Min.**		**Average**		**Max.**	
New-Order	18.10	0.02	18.11	12.05	18.29	120.28
Payment	3.04	0.02	3.06	12.08	3.26	120.36
Order-Status	2.04	0.02	2.06	10.19	2.23	101.75
Delivery (interactive)	2.04	0.02	2.06	5.07	2.22	49.81
Stock-Level	2.04	0.01	2.06	5.06	2.20	48.98
Test Duration						
Ramp-up time (seconds)					**2,130**	
Measurement interval					**1,800**	
Transactions during measurement interval					**993,408**	
Checkpointing						
Number of checkpoints					**1**	
Checkpoint interval					**1,800**	
Reproducibility Run						
Reported measurement					**33,113.60**	
Reproducibility measurement					**33,043.93**	
Difference					**0.21%**	

Table Of Contents

General Items

Application Code and Definition Statements

The application program (as defined in clause 2.1.7) must be disclosed. This includes, but is not limited to, the code implementing the five transactions and the terminal input output functions.

Appendix A and B contain all source code implemented in this benchmark.

Test Sponsor

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

Fujitsu sponsored and conducted this TPC Benchmark C.

Parameter Settings

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

- *Database options,*
- *Recover/commit options,*
- *Consistency/locking options*
- *Operating system and application configuration parameter.*

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

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

Configuration Items

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

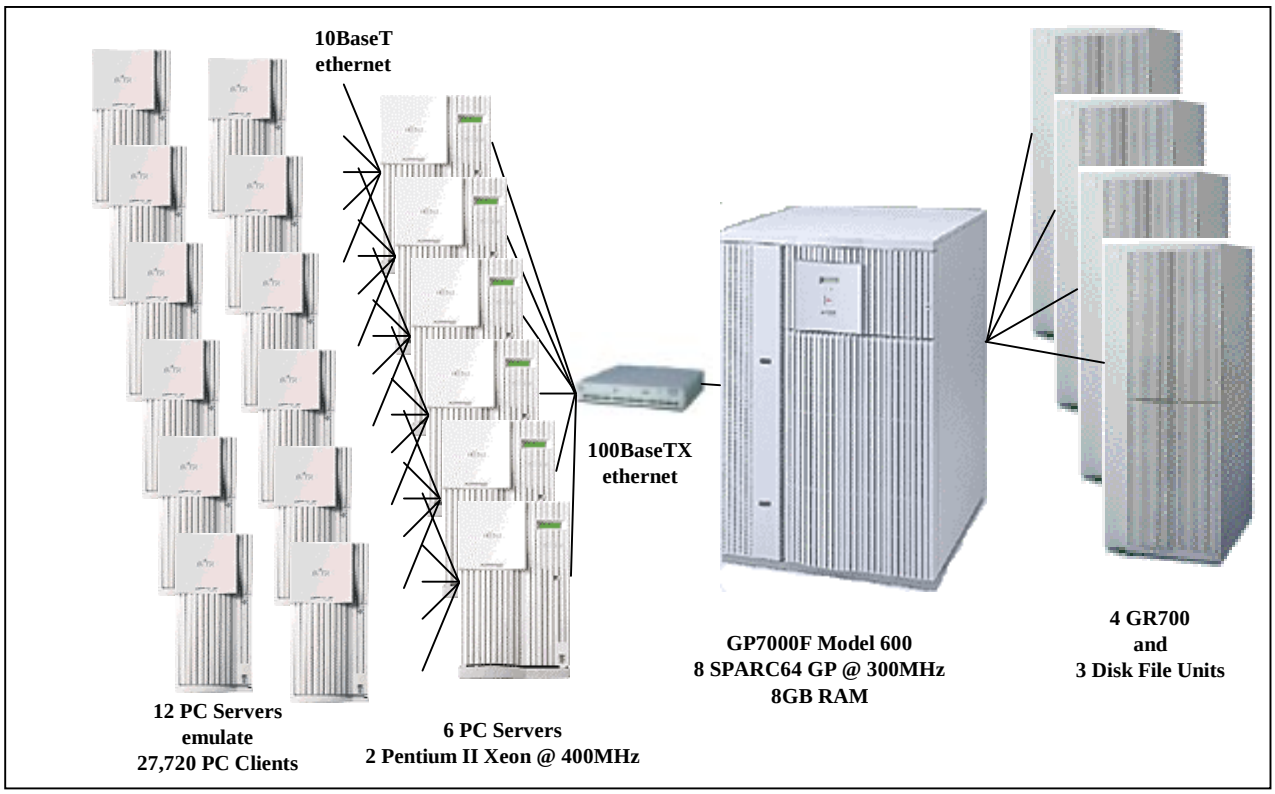
The System Under Test (SUT), a GP7000F Model 600, is depicted in the following diagrams.

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

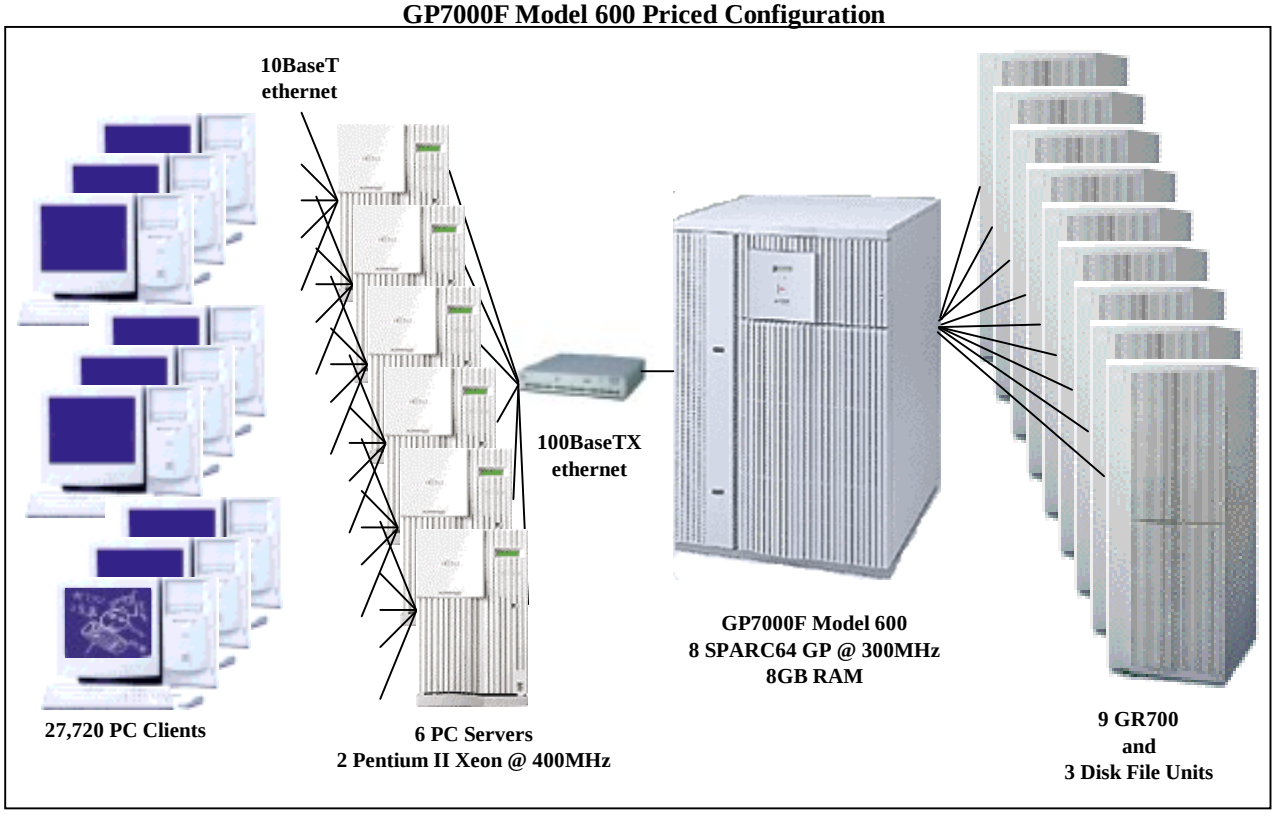
The configuration diagrams for both the tested and priced systems are included on the following pages.

The only difference is the use of the RTE.

GP7000F Model 600 Tested Configuration



GP7000F Model 600 Priced Configuration



Clause 1 Related Items

1.1 Table Definitions

Listings must be provided for all table definition statements and all other statements used to set up the database.

Appendix E contains the code used to define and load the database tables.

1.2 Physical Organization of Database

The physical organization of tables and indices within the database must be disclosed.

Appendix G discloses the organization of tables and indices on the disks.

1.3 Insert and Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restrictions in the SUT database implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the maximum key value for these new rows.

All insert and delete functions were verified and fully operational during the entire benchmark.

1.4 Partitioning

While there are a few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C benchmark, any such partitioning must be disclosed.

All tables were horizontally partitioned except for Items. Each table was horizontally partitioned following the w-id values given below:

Table	DSI (Data Structure Instance)
Warehouse	66 WH
District	66 WH
Customer	11 WH
History	11 WH
Order	11 WH
New Order	11 WH
Order Line	11 WH
Stock	33 WH

1.5 Replication, Duplication or Additions

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

No replications, duplications or additional attributes were used in this benchmark.

Clause 2 Related Items

2.1 Random Number Generation

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

The seeds for each user were generated using the process id. Each RTE machine was given a number incremented by 30,000. The process id was appended to this number to ensure uniqueness across all RTE machines. These seeds were printed to a file and verified by the auditor to be unique.

2.2 Input/Output Screen Layout

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

All screen layouts followed the specification exactly.

2.3 Priced Terminal Feature Verification

The method used to verify that the emulated terminals provide all the features described in Clause 2.2.2.4 must be explained. Although not specifically priced, the type and model of the terminals used for the demonstration in 8.1.3.3 must be disclosed and commercially available (including supporting software and maintenance).

The terminal attributes were verified by the auditor manually exercising each specification during the onsite audit portion of this benchmark.

2.4 Presentation Manager or Intelligent Terminal

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

Application code running on the client machines implemented the TPC-C user interface. No presentation manager software or intelligent terminal features were used. The source code for the forms applications is listed in Appendix A.

2.5 Transaction Statistics

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

Table 2. 1 Transaction Statistics

Statistic		Value
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	1.02%
	Average items per order	10.00
Payment	Home warehouse	84.97%
	Remote warehouse	15.03%
	Accessed by last name	60.02%
Order Status	Accessed by last name	59.84%
Delivery	Skipped transactions	None
Transaction Mix	New Order	44.79%
	Payment	43.08%
	Order status	4.06%
	Delivery	4.03%
	Stock level	4.04%

2.6 Queueing Mechanism

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

Delivery transactions were submitted to servers using the same mechanism that other transactions used. The only difference was that the Tuxedo call to the server process was asynchronous, i.e., control would return to the client process immediately and the deferred delivery part would complete asynchronously on the server.

Clause 3 Related Items

3.1 Transaction System Properties (ACID)

The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met. This includes disclosing which case was followed for the execution of Isolation Test 7.

The TPC Benchmark C Standard Specification defines a set of transaction processing system properties that a SUT must support during the execution of the benchmark. Those properties are Atomicity, Consistency, Isolation and Durability (ACID).

This section defines each of those properties, describes the steps taken to ensure that they were present during the test and describes a series of tests done to demonstrate compliance with the specification.

3.2 Atomicity

The system under test must guarantee that the database transactions are atomic; the system will either perform all individual operations on the data or will assure that no partially completed operations leave any effects on the data.

3.2.1 Completed Transactions

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number as specified in Clause 2.5.1.2) and verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have been changed appropriately.

A row was randomly selected from the warehouse, district and customer tables, and the balances noted. A payment transaction was started with the same warehouse, district and customer identifiers and a known amount. The payment transaction was committed and the rows were verified to contain correctly updated balances.

3.2.2 Aborted Transactions

Perform the Payment transaction for a randomly selected warehouse, district and customer (by customer number as specified in Clause 2.5.1.2) and substitute a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have NOT been changed.

A row was randomly selected from the warehouse, district and customer tables, and the balances noted. A payment transaction was started with the same warehouse, district and customer identifiers and a known amount. The payment transaction was rolled back and the rows were verified to contain the original balances.

3.3 Consistency

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

The benchmark specification requires explicit demonstration of the following four consistency conditions;

- The sum of the district balances in a warehouse is equal to the warehouse balance;
- for each district, the next order id minus one is equal to the maximum order id in the ORDER table and equal to the maximum new order id in the NEW-ORDER table;
- for each district, the maximum order id minus minimum order id in the ORDER table plus one equals the number of rows in the NEW-ORDER table for that district;
- for each district, the sum of the order line counts in the ORDER table equals the number of rows in the ORDER-LINE table for that district.

These consistency conditions were tested using a shell script to issue queries to the database. The results of the queries verified that the database was consistent for all four tests.

A performance run was completed including a full 30 minutes of steady state and checkpoints.

The shell script was executed again. The result of the same queries verified that the database remained consistent after the run.

3.4 Isolation

Isolation can be defined in terms of phenomena that can occur during the execution of concurrent transactions. These phenomena are P0 ("Dirty Write"), P1 ("Dirty Read"), P2 ("non-repeatable Read"), and P3 ("Phantom"). The table in Clause 3.4.1 of the TPC-C specifications defines the isolation requirements which must be met by the TPC-C transactions. Sufficient conditions must be enabled at either the system or application level to ensure the required isolation defined above (clause 3.4.1) is

obtained.

The benchmark specification defines nine required tests to be performed to demonstrate that the required levels of transaction isolation are met. These tests, described in Clauses 3.4.2.1 - 3.4.2.9, were all performed and verified as required.

Isolation tests one through nine were executed using shell scripts to issue queries to the database. Each script included timestamps to demonstrate the concurrency of operations. The results of the queries were captured to files. The captured files were verified by the auditor to demonstrate the required isolation had been met.

For Isolation test seven, case A was followed.

3.5 Durability

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

3.5.1 Durable Media Failure

3.5.1.1 Loss of Log And Data

To demonstrate recovery from a permanent failure of durable medial containing the SymfoWARE recovery log data and TPC-C tables, the following steps were executed on a database of 2,640 warehouses:

1. The database was backed up to extra disks.
2. The total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
3. The RTEs were started with 26,400 users.
4. The test was allowed to run for a minimum of 5 minutes.
5. One of the log disks was powered off by removing it from the cabinet. Since the log was mirrored, the transactions continued to run without interruption.
6. The test was allowed to run for another 5 minutes and a disk array failure was caused by removing a disk from the disk array cabinet.
7. The RTEs were shut down.
8. A new disk was inserted into the disk cabinet and the data disk was reformatted to simulate a complete loss of data.
9. SymfoWARE was restarted.
10. Data from the backup disk was copied to the new disk and SymfoWARE used the transaction logs to roll forward the recovery data from committed transactions.
11. Step 2 was repeated and the difference between the first and second counts noted.
12. The success file was used to determine the number of NEW_ORDERS successfully returned to the RTEs.
13. The counts in step 11 and 12 were compared, and the results verified that all committed transactions were successfully recovered.
14. Data from the success file was used to query the database to demonstrate that successful transactions had corresponding rows in the ORDER table and that rolled back transactions did not.

3.5.2 Instantaneous Interruption and Loss of Memory

Because loss of power erases the contents of memory, the instantaneous interruption and the loss of memory tests were combined into a single test. This test was executed on a fully scaled database of 2,640 warehouses under a full load of 26,400 users. The following steps were executed:

1. The total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving the beginning count.
2. The RTE was started with 26,400 users.
3. The test was allowed to run for a minimum of 20 minutes.
4. A checkpoint was enforced.
5. The test was allowed to run for another minute.
6. The primary power to the processor was shutdown.
7. The RTE was shutdown.
8. Power was restored and the system performed an automatic recovery.
9. SymfoWARE was restarted and performed an automatic recovery.
10. Step 1 was repeated and the difference between the first and second counts was noted.
11. The success file was used to determine the number of NEW-ORDERS successfully returned to the RTE.
12. The counts in step 10 and 11 were compared and the results verified that all committed transactions had been successfully recovered.
13. Data from the success file was used to query the database to demonstrate successful transactions had corresponding rows in the ORDER table, and rolled back transactions did not.

Clause 4 Related Items

4.1 Initial Cardinality of Tables

The cardinality (e.g. number of rows) of each table, as it existed at the start of the benchmark run, must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table were deleted, the cardinality of the WAREHOUSE table as initially configured and the number of rows deleted must be disclosed.

The TPC-C database was initially configured with 2,772 warehouses.

Table 4.1 Number of Rows for Server

Table	Occurrences
Warehouse	2,772
District	27,720
Customer	83,160,000
History	83,160,000
Order	83,160,000
New Order	24,948,000
Order Line	831,639,940
Stock	277,200,000
Item	100,000

4.2 Database Layout

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

Section 1.2 of this report details the distribution of database tables across all disks. The code that creates the tables is included in Appendix E.

4.3 Type of Database

A statement must be provided that describes:

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

SymfoWARE is a relational DBMS.

The interface used was SymfoWARE stored procedures embedded in C code. The new-order transaction also used COBOL to accomplish bulk inserts of the order lines.

4.4 Database Mapping

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

The database, with the exception of the Item table, was horizontally partitioned. This partitioning is fully described in Section1.4.

4.5 180 Day Space

Details of the 180 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.

The 180 day space requirement is shown in Appendix F.

The archive log grows at the rate of 6.3732KB per New-Order transaction, which was measured from the steady state. The 8 hours log space was 96.61GB at the measured rate and 161.30GB of log space was prepared for the measurement.

For dynamic tables the following steps were followed:

1. The number of rows and number of used blocks were counted on a freshly loaded database.
2. The number of rows was divided by the number of blocks, giving rows per block.
3. The number of rows inserted in 8 hours was estimated equal to tpmC for HISTORY and ORDER, and ten times tpmC for ORDERLINE.
4. The number of rows in step 3 was divided by the number derived in step 2.
5. The number in step 4 was added to the number of used blocks from step 1.
6. The database was queried to show the space allocated exceeded the number in step 5.

Clause 5 Related Items

5.1 Throughput

Measured tpmC must be reported.

Measured tpmC	33,113.60
Price per tpmC	$83.74

5.2 Response Times

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

An emulation delay of 0.1 second is included in response time and menu time to compensate for browser delay.

Table 5.1 Response Times

Type	Average	Maximum	90th %
New-Order	1.26	66.23	1.60
Payment	0.83	66.33	1.48
Order-Status	0.80	61.07	1.46
Interactive Delivery	0.23	49.36	0.16
Deferred Delivery	1.09	61.80	1.86
Stock-Level	0.81	61.09	1.49
Menu	0.40	50.24	0.15

5.3 Keying and Think Times

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

Table 5.2 Keying Times

Type	Minimum	Average	Maximum
New-Order	18.10	18.11	18.29
Payment	3.04	3.06	3.26
Order-Status	2.04	2.06	2.23
Interactive Delivery	2.04	2.06	2.22
Stock-Level	2.04	2.06	2.20

Table 5.3 Think Times

Type	Minimum	Average	Maximum
New-Order	0.02	12.05	120.28
Payment	0.02	12.08	120.36
Order-Status	0.02	10.19	101.75
Interactive Delivery	0.02	5.07	49.81
Stock-Level	0.01	5.06	48.98

5.4 Response Time Frequency Distribution Curves and Other Graphs

Response Time frequency distribution curves (see Clause 5.6.1) must be reported for each transaction type.

The performance curve for response times versus throughput (see Clause 5.6.2) must be reported for the New-Order transaction.

Think Time frequency distribution curves (see Clause 5.6.3) must be reported for the New-Order transaction.

A graph of throughput versus elapsed time (see Clause 5.6.5) must be reported for the New-Order transaction.

Figure 5.1: New Order Response Time Distribution

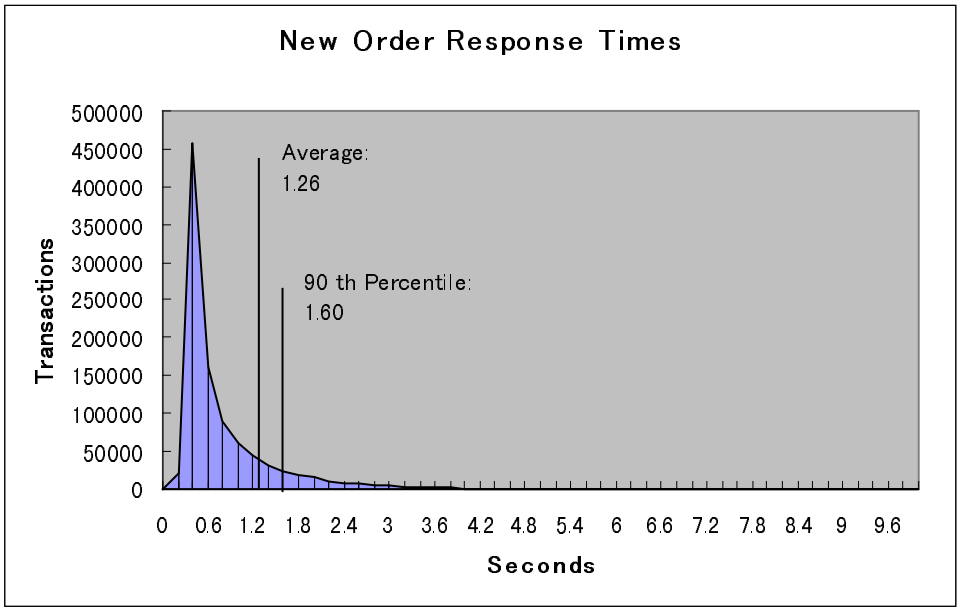


Figure 5.2: Payment Response Time Distribution

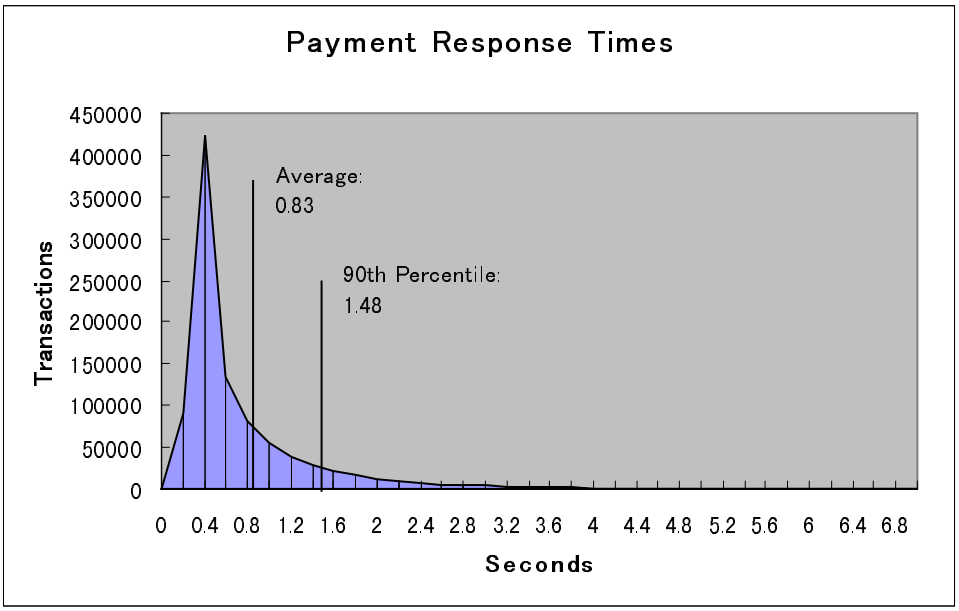


Figure 5.3: Order Status Response Time Distribution

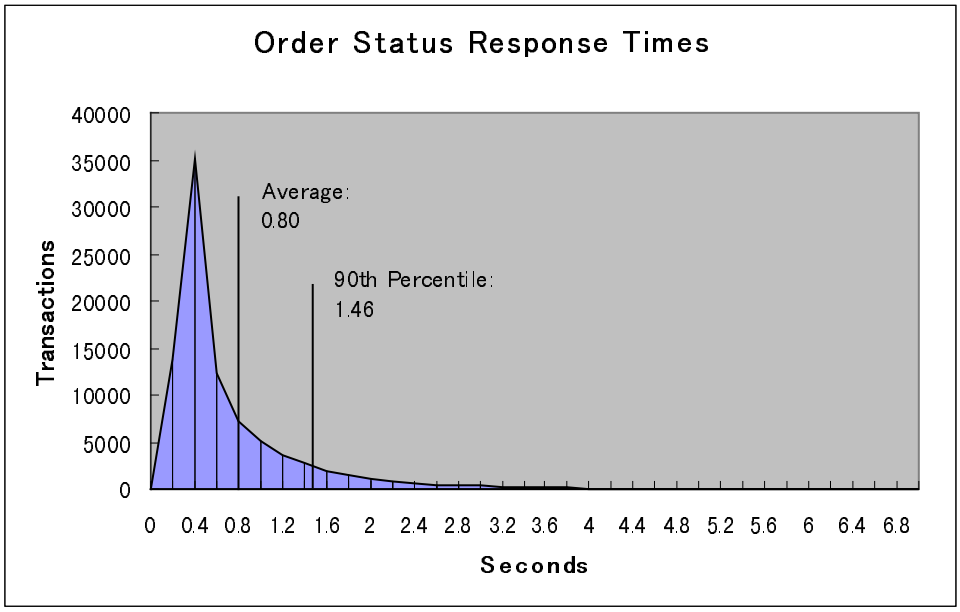


Figure 5.4: Delivery Response Time Distribution

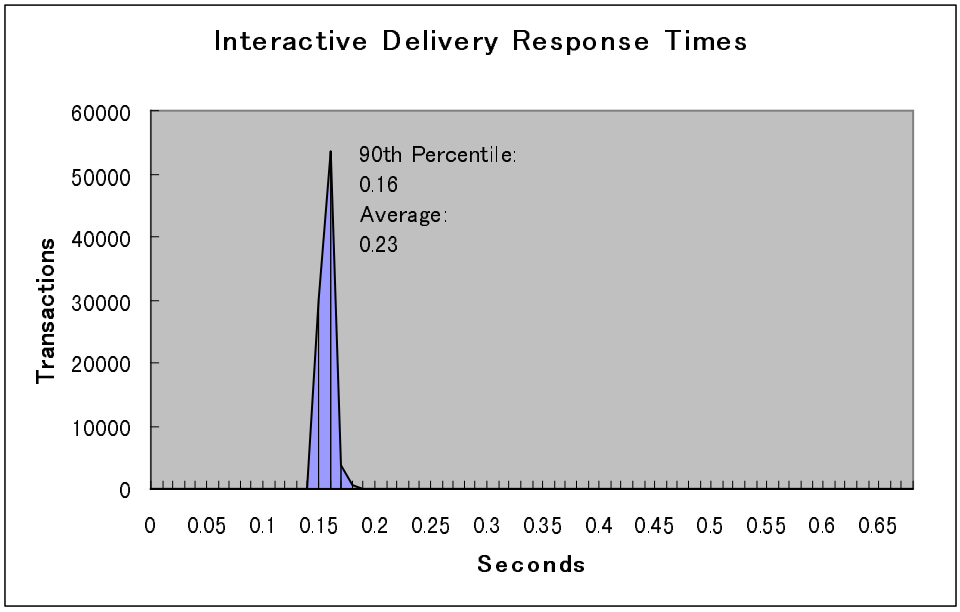


Figure 5.5: Stock Level Response Time Distribution

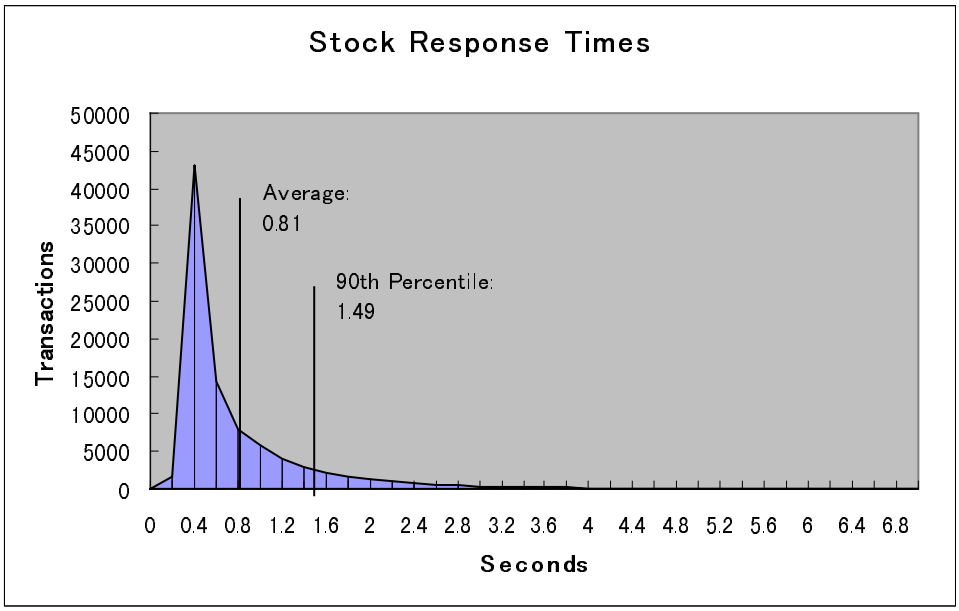


Figure 5.6: New Order Think Time Frequency Distribution

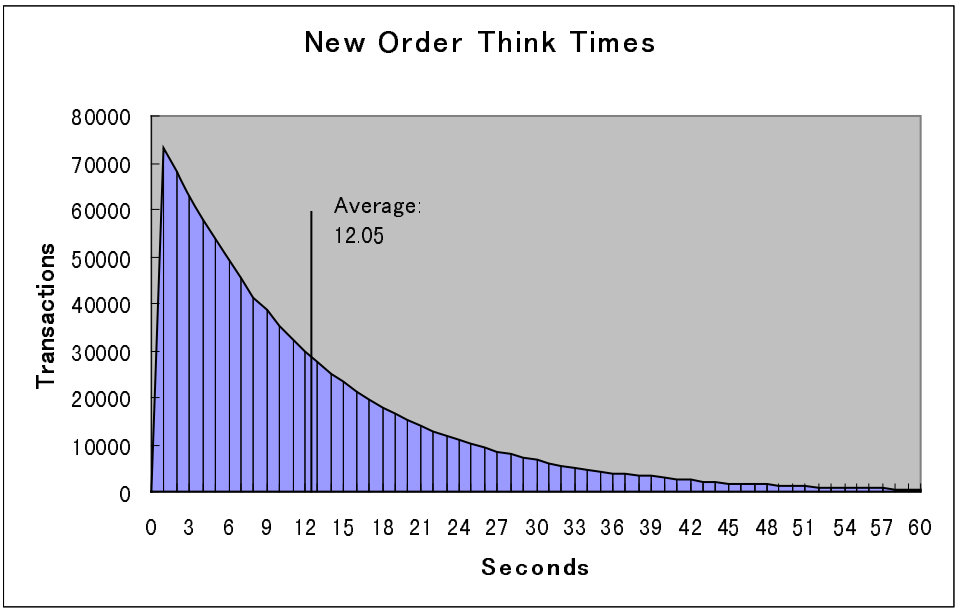


Figure 5.7: Response time versus Throughput

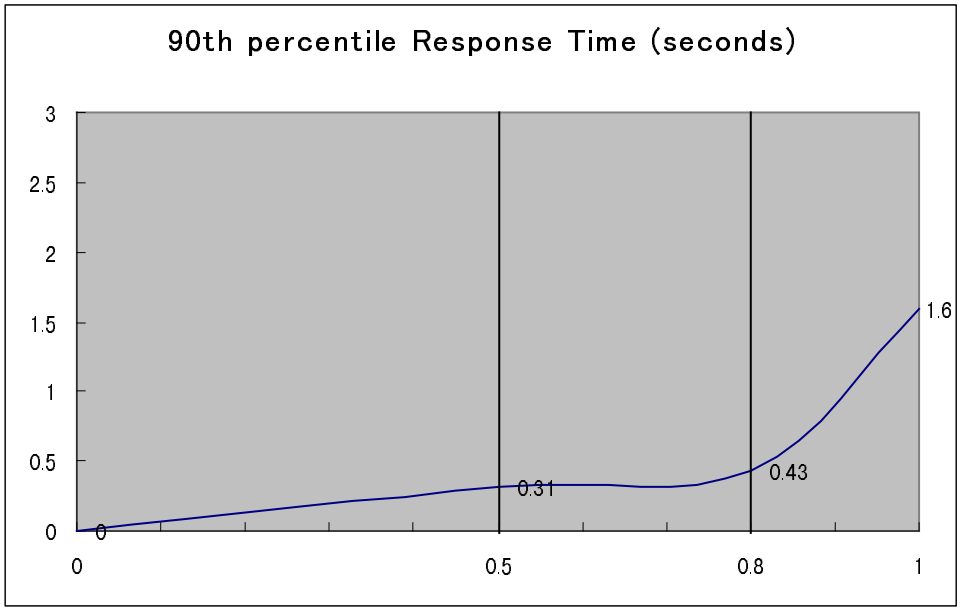
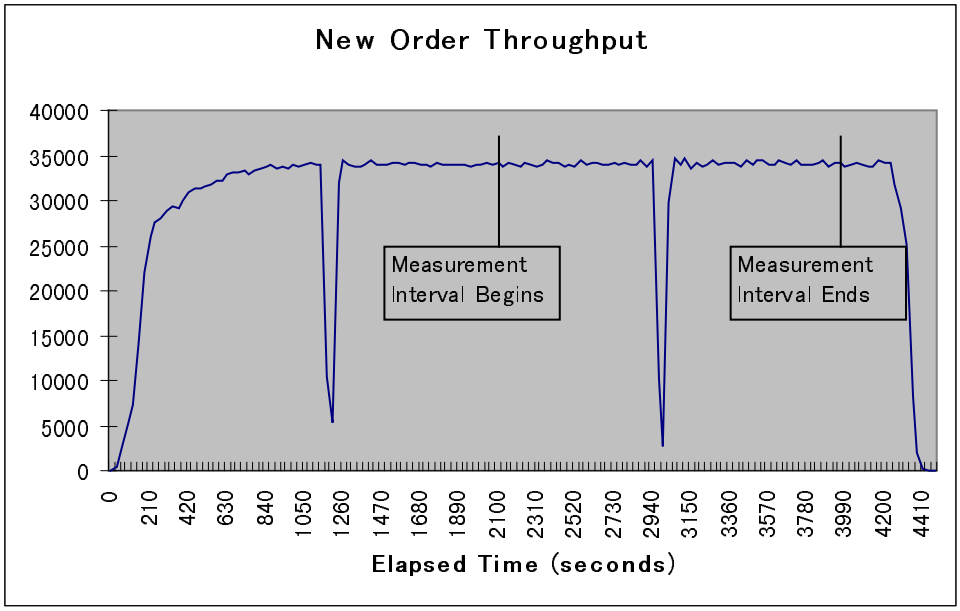


Figure 5.8: New Order Sustained Throughput



5.5 Steady State Determination

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

Steady state was determined by examining data reported for each 30-second interval over the duration of the measured run. Steady state was further confirmed by the throughput data collected during the run and graphed in Figure 5.8.

5.6 Work Performed During Steady State

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

A SymfoWARE checkpoint forces all "dirty" pages (pages that have been updated since they were last written) to be physically written to the durable disks SymfoWARE executes a checkpoint for the following conditions:

1. The amount of recovery data reaches the value specified at the creation of the temporary log, which contains the before images and after images of each transaction. The interval the recovery data takes to reach the specified value depends upon workload. The temporary log is configured by the *rdblog* command.
2. Upon an explicit *rdbrcp* request.

For each benchmark measurement, after all users are active, the script that issues *rdbrcp* is started manually on the server. The script sleeps and performs another checkpoint every 30 minutes, which is equal to the measurement interval. *Rdbrcp* notifies the time upon the completion of the checkpoint and the start time and end time of all checkpoints are captured to a flat file. The recovery log is configured to be large enough that no other checkpoint will occur during the measurement. The recovery log is marked as reusable after the checkpoint completes. The positioning of the checkpoint is verified to be clear of the guard zones and is depicted on the graph in Figure 5.8.

5.7 Reproducibility

A description of the method used to determine the reproducibility of the measurement results must be reported.

The measurement procedure was repeated and the throughput verified to be within less than 2% of the reported measurement.

5.8 Measurement Period Duration

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

The reported measured interval was exactly 30 minutes long.

5.9 Regulation of Transaction Mix

The method of regulation of the transaction mix (e.g., card decks or weighted random distribution) must be described. If weighted distribution is used and the RTE adjusts the

weights associated with each transaction type, the maximum adjustments to the weight from the initial value must be disclosed.

The RTE used the UNIX function lrand48() to control the transaction mix, and could not be adjusted during the run.

5.10 Transaction Statistics

The percentage of the total mix for each transaction type must be disclosed. The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed. The average number of order-lines entered per New-Order transaction must be disclosed. The percentage of remote order lines per New-Order transaction must be disclosed. The percentage of remote Payment transactions must be disclosed. The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed. The percentage of Delivery transactions skipped due to there being fewer than necessary orders in the New-Order table must be disclosed.

Table 5.4: Transaction Statistics

Statistics		Value
Transaction Mix	New Order	44.79%
	Payment	43.08%
	Order status	4.06%
	Delivery	4.03%
	Stock level	4.04%
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	1.02%
	Average items per order	10.00
Payment	Home warehouse	84.97%
	Remote warehouse	15.03%
	Accessed by last name	60.02%
Order Status	Accessed by last name	59.84%
Delivery	Skipped transactions	None

5.11 Checkpoint Count and Location

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

One checkpoint was recorded before the measured window opened and another checkpoint was started 860 seconds inside the measured window. Both checkpoints were clear of the guard zone. Checkpoints were started exactly 30 minutes apart.

Clause 6 Related Items

6.1 RTE Descriptions

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

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

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

6.2 Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to the priced system. The results of the test described in Clause 6.6.3.4 must be disclosed.

There were no emulated components in the benchmark configuration other than the emulated users' workstations.

6.3 Functional Diagrams

A complete functional diagram of both the benchmark configuration and the configuration of the proposed (target) system must be disclosed. A detailed list of all hardware and software functionality being performed on the Driver System and its interface to the SUT must be disclosed.

The driver system performed the data generation and input functions of the display device. It also captured the input and output data and timestamps for post-processing of the reported metrics. No other functionality was included on the driver system

The abstract at the beginning of this report contains detailed diagrams of both the benchmark configuration and the priced configuration, including the driver system.

6.4 Networks

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

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

A 100Mbps ethernet LAN connection was used between each client and the server. Fifteen 10Mbps ethernet LAN connections were used between the emulated users and the client machines.

6.5 Operator Intervention

If the configuration requires operator intervention (see Clause 6.6.6), the mechanism and the frequency of this intervention must be disclosed.

This configuration does not require any operator intervention to sustain eight hours of the reported throughput, other than beginning the checkpointing process.

Clause 7 Related Items

7.1 System Pricing

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

The total 5 year price of the entire configuration must be reported, including: hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

A detailed price list is included in the abstract at the beginning of this report.

7.2 Availability

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

All hardware and software components will be available no later than December 23, 1999.

7.3 Throughput and Price Performance

A statement of the measured tpmC as well as the respective calculations for the 5-year pricing, price/performance (price/tpmC), and the availability date must be included.

Maximum Qualified Throughput:	33,113.60
Price per tpmC	$83.74
Available	December 23, 1999

7.4 Country Specific Pricing

Additional Clause 7 related items may be included in the Full Disclosure Report for each country specific priced configuration. Country specific pricing is subject to Clause 7.1.7

This system is being priced for the United States of America.

7.5 Usage Pricing

For any usage pricing, the sponsor must disclose:

- *Usage level at which the component was priced.*
- *A statement of the company policy allowing such pricing.*

SymfoWARE is sold with a 252 user license. There were 252 connections between the clients and server.

Clause 9 Related Items

9.1 Auditor's Report

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

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

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

9.2 Availability of the Full Disclosure Report

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

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

Transaction Processing Performance Council
c/o Shanley Public Relations
777 North First Street, Suite 6000
San Jose, CA 95112-6311
408/295-8894

Appendix A: Client Source Code

File: bench2.h'

```
/*
        bench2.h : Data structure for message
send/receive

        Version     Beta  1995/02/24
        Version     Beta2 1995/03/06
        Version     Beta2a      1995/03/14
        Version     Beta3 1995/03/23
        Version     1.0   1998/02/24 for Solaris 2.x
*/


typedef struct {
        int     tx_type;
        int     C_R;

        int     errorpos;           /* 1997.03.13 */
        int     sqlstate;           /* 1997.03.13 */

        short   w_id;

        short   d_id;

        short   o_carrier_id;

        long    startsec;
        long    startusec;
} delivery_trans;


typedef struct {
        int     tx_type;
        int     C_R;

        int     errorpos;           /* 1997.03.13 */
        int     sqlstate;           /* 1997.03.13 */

        long    threshold;
        long    low_stock;

        short   w_id;

        short   d_id;
} stocklvl_trans;


typedef struct {
        int     tx_type;
        int     C_R;

        int     errorpos;           /* 1997.03.13 */
        int     sqlstate;           /* 1997.03.13 */

        short   w_id;
        char    w_street_1[21];
        char    w_street_2[21];
        char    w_city[21];
        char    w_state[3];
        char    w_zip[10];

        short   d_id;
        char    d_street_1[21];
        char    d_street_2[21];
        char    d_city[21];
        char    d_state[3];
        char    d_zip[10];

/*      short   c_id;*/
        int     c_id;
        short   c_d_id;
        short   c_w_id;
        char    c_first[17];
        char    c_middle[3];
        char    c_last[17];
        char    c_street_1[21];
        char    c_street_2[21];
        char    c_city[21];
        char    c_state[3];
        char    c_zip[10];
        char    c_phone[17];
        double      c_since;
        char    c_credit[3];
        double      c_credit_lim;
        /*long  c_credit_lim;*/
        long    c_discount;
        double      c_balance;
        /*long  c_balance;*/
        char    c_data[501];

        double      h_date;
        long    h_amount;
} payment_trans;


typedef struct {
        int     tx_type;
        int     C_R;

        int     errorpos;           /* 1997.03.13 */
        int     sqlstate;           /* 1997.03.13 */

        short   w_id;

        short   d_id;

/*      short   c_id;*/
        int     c_id;
        char    c_first[17];
        char    c_middle[3];
        char    c_last[17];
        double      c_balance;
        /*long  c_balance;*/

        long    o_id;
        double      o_entry_d;
        short   o_carrier_id;
        short   o_ol_cnt;

        long    ol_i_id[15];
        short   ol_supply_w_id[15];
        double      ol_delivery_d[15];
        short   ol_quantity[15];
        long    ol_amount[15];
        /*double    ol_amount[15];*/
} orderstat_trans;


typedef struct {
        int     tx_type;
        int     C_R;

        int     errorpos;           /* 1997.03.13 */
        int     sqlstate;           /* 1997.03.13 */

        char    brand_generic[15];
        long    i_price[15];
        /*double    i_price[15];*/
        char    i_name[15][25];
        long    total_amount;
        /*double    total_amount;*/

        short   w_id;
        long    w_tax;

        short   d_id;
        long    d_tax;

/*      short   c_id;*/
        int     c_id;
        char    c_last[17];
        char    c_credit[3];
        long    c_discount;

        long    o_id;
        double      o_entry_d;
        short   o_ol_cnt;

        long    ol_i_id[15];
        short   ol_supply_w_id[15];
        short   ol_quantity[15];
        long    ol_amount[15];
        /*double    ol_amount[15];*/

        long    s_quantity[15];
} neworder_trans;


#if 0

typedef struct {
        int     tx_type;
        int     C_R;
        long    threshold;
        long    low_stock;
        char    brand_generic[15];
        long    i_price[15];
        /*double    i_price[15];*/
        char    i_name[15][25];
        long    total_amount;
        /*double    total_amount;*/
        double      pl_delivery_d[15];

        short   w_id;
        char    w_name[11];
        char    w_street_1[21];
        char    w_street_2[21];
        char    w_city[21];
        char    w_state[3];
        char    w_zip[10];
        long    w_tax;
        double      w_ytd;

        short   d_id;
```

```
        char  d_name[11];
        char  d_street_1[21];
        char  d_street_2[21];
        char  d_city[21];
        char  d_state[3];
        char  d_zip[10];
        long  d_tax;
        long  d_next_o_id;

/*      short c_id;*/
        int     c_id;
        short c_d_id;
        short c_w_id;
        char  c_first[17];
        char  c_middle[3];
        char  c_last[17];
        char  c_street_1[21];
        char  c_street_2[21];
        char  c_city[21];
        char  c_state[3];
        char  c_zip[10];
        char  c_phone[17];
        double        c_since;
        char  c_credit[3];
        double        c_credit_lim;
        /*long c_credit_lim;*/
        long  c_discount;
        double        c_balance;
        /*long c_balance;*/
        double        c_ytd_payment;
        short c_payment_cnt;
        /*long c_payment_cnt;*/
        char  c_data[501];

        double        h_date;
        long  h_amount;
        char  h_data[25];

        long  no_o_id;

        long  o_id;
        double        o_entry_d;
        short o_carrier_id;
        short o_ol_cnt;
        short o_all_local;

        long  ol_number;
        long  ol_i_id[15];
        short ol_supply_w_id[15];
        double        ol_delivery_d[15];
        short ol_quantity[15];
        long  ol_amount[15];
        /*double      ol_amount[15];*/
        char  ol_dist_info[24];

        long  s_quantity[15];
        char  s_dist_01[24];
        char  s_dist_02[24];
        char  s_dist_03[24];
        char  s_dist_04[24];
        char  s_dist_05[24];
        char  s_dist_06[24];
        char  s_dist_07[24];
        char  s_dist_08[24];
        char  s_dist_09[24];
        char  s_dist_10[24];
        double        s_ytd;
        long  s_order_cnt;
        long  s_remote_cnt;
        char  s_data[51];
} trans_buf;


main()
{
        printf( "%d %d %d %d %d %d\n",
                sizeof( delivery_trans ),
                sizeof( stocklvl_trans ),
                sizeof( payment_trans ),
                sizeof( orderstat_trans ),
                sizeof( neworder_trans ),
                sizeof( trans_buf ) );
        return 0;
}

#endif
```

File: dbgprt.h

```
#ifdef SCRTEST

// Prottype of Debug Print Function

extern  "C" void oder_dsp(rte_input_data *,
orderstat_trans *, int, int);
extern  "C" void pay_dsp (rte_input_data *,
payment_trans *, int, int);
extern  "C" void sto_dsp (rte_input_data *,
stocklvl_trans *, int, int, int);
extern  "C" void new_dsp (rte_input_data *,
neworder_trans *, int, int, int);

extern  "C" void dummy_delivery ( delivery_trans * );
extern  "C" void dummy_stocklvl ( stocklvl_trans * );
extern  "C" void dummy_orderstat( orderstat_trans * );
extern  "C" void dummy_payment  ( payment_trans * );
extern  "C" void dummy_neworder ( neworder_trans * );
extern  "C" char *get_datetimestr( char * );
extern  "C" char *get_datestr   ( char * );

#endif
```

File: delpage.h

```
/* ---------------------------------------------------------------------
-
   delpage.h
   data of delivery transaction result screen (HTML
format)
   ---------------------------------------------------------------------
- */

/* Header data */
#define h_del1 "\
<HTML><HEAD><TITLE>TPC-WINDOW
</TITLE></HEAD><BODY>\n\n\
<CENTER>Delivery<br></CENTER>\
<font size=4>\r\n<PRE>"

/* Screen data*/
#define h_del2 "\
Warehouse:     \r\n\
\r\n\
Carrier Number:   \r\n\
\r\n\
Execution Status:                       \r\n\
\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\
"
/* Tailer data */
#define h_del3 "\
</PRE><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"New
order\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"Order
Status\">\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"Stock
Level\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</FORM></BODY></HTML>\n"

/* Offset to field which should set data */
int  delp[] = { 0xb, 0x23, 0x3b }; /* w_id, carrier number,
status */
```

File: dmy.h

```
//
// dmy.h
//

#ifdef SCRTEST

char *get_datetimestr( char *buf )
{
        struct tm     *tm;
        time_t                tim;

        time( &tim );
        tm  = localtime( &tim );

        sprintf( buf, "%2d-%2d-%4d %2d:%2d:%2d", tm-
>tm_mday, tm->tm_mon+1,
                tm->tm_year+1900, tm->tm_hour, tm-
>tm_min, tm->tm_sec );

        return buf;
}


char *get_datestr( char *buf )
{
        struct tm     *tm;
        time_t                tim;

        time( &tim );
        tm  = localtime( &tim );

        sprintf( buf, "%2d-%2d-%4d",
                tm->tm_mday, tm->tm_mon+1, tm-
>tm_year+1900 );
        return buf;
}


void dummy_delivery( delivery_trans *bp )
{
```

```
#ifdef Symfo
        bp->C_R = 1;
#else
        bp->C_R = NOERR;
#endif

        return;
}


void dummy_stocklvl( stocklvl_trans *bp )
{
        int     i;

#ifdef Symfo
        bp->C_R = 1;
#else
        bp->C_R = NOERR;
#endif

        do{
                i = rand()%1000;
        } while ( i > bp->threshold );

        bp->low_stock = i;

        return;
}


void dummy_payment( payment_trans *bp )
{

#ifdef Symfo
        bp->C_R = 1;
#else
        bp->C_R = NOERR;
#endif

//      get_datetimestr( bp->h_date );
                // check
        strcpy( bp->w_street_1, "Baker street" );
        strcpy( bp->w_street_2, "221B" );
        strcpy( bp->w_city, "London" );
        strcpy( bp->w_state, "GB" );
        strcpy( bp->w_zip, "88033000" );

        strcpy( bp->d_street_1, "Minato-ku" );
        strcpy( bp->d_street_2, "Azabu 10" );
        strcpy( bp->d_city, "Tokyo" );
        strcpy( bp->d_state, "JP" );
        strcpy( bp->d_zip, "102" );

        bp->c_id = 777;
        strcpy( bp->c_first, "John" );
        strcpy( bp->c_middle, "H" );
        strcpy( bp->c_last, "Watson" );
        strcpy( bp->c_street_1, "Baker street" );
        strcpy( bp->c_street_2, "221B" );
        strcpy( bp->c_credit, "GC" );

#ifdef Symfo
        bp->c_discount = 20;
#else
        bp->c_discount = (float)0.20;
        // check
#endif
        strcpy( bp->c_city, "London" );
        strcpy( bp->c_state, "GB" );
        strcpy( bp->c_zip, "888" );
        strcpy( bp->c_phone, "1234567890123456");
        bp->c_balance = 67876;
        bp->c_credit_lim = 77777;
//      get_datestr( bp->c_since );
                // check

        strcpy( bp->c_data,
"Migyamigyamigyamigyamigya"
                        "migyamigyamigyamigyamigya" );
        return;
}


void dummy_orderstat( orderstat_trans *bp )
{
        int     i, j;

#ifdef Symfo
        bp->C_R = 1;
#else
        bp->C_R = NOERR;
#endif

        bp->c_id = rand()%10000;
        strcpy( bp->c_first, "Robert" );
        strcpy( bp->c_middle, "L" );
        strcpy( bp->c_last, "Fish" );
        bp->c_balance = ( ( rand()*rand()%19999999 )-
9999999 ) / (double)100.0;
        /*
        fprintf( stderr, "ordout.c_balance = %12.4f\n", bp-
>ordout.c_balance );
        bp->c_balance = -1;
        */

        bp->o_id = rand()%10000;
//      get_datetimestr( bp->o_entry_d );
        // check
        bp->o_carrier_id = rand()%100;

        bp->o_ol_cnt = ( rand()%11 )+5;
        j = bp->o_ol_cnt;
        for ( i = 0; i < j; i++ )
        {
                bp->ol_supply_w_id[i] = ( rand()%10 )+1;
                bp->ol_i_id[i] = ( rand()%100000 )+1;
                bp->ol_quantity[i] = ( rand()%99 )+1;

#ifdef Symfo
                bp->ol_amount[i] = (long)((float)rand() *
(float)100);  // check
#else
                bp->ol_amount[i] = (float)rand();
        // check
#endif

//              debug2( ( stderr, "rand : %f\n", bp-
>ordout.ol_amount[i] ) );

//              get_datetimestr( bp->ol_delivery_d[i] );
        // check
        }

        return;
}


void dummy_neworder( neworder_trans *bp )
{
        static int      o_id = 3001;
        int i;


#ifdef Symfo
        bp->C_R = 1;
#else
        bp->C_R = NOERR;
#endif

        //      *( bp->status ) = '\0';
        // This value is nothing

        strcpy( bp->c_last, "Holmes" );
        strcpy( bp->c_credit, "GC" );
        bp->o_id = o_id++;
/*
        bp->newout.o_id = ( rand()%100000 )+1;
*/

//      get_datetimestr( bp->o_entry_d );               //
check

#ifdef Symfo
        bp->c_discount = (long)(rand()%101 );       //
check
        bp->w_tax = (long)( rand()%2001 );          //
check
        bp->d_tax = (long)( rand()%2001 );          //
check
#else
        bp->c_discount = (float)(( rand()%101 )/10000.0);
        // check
        bp->w_tax = (float)(( rand()%2001 )/10000.0);
        // check
        bp->d_tax = (float)(( rand()%2001 )/10000.0);
        // check
#endif

        bp->total_amount = 0;
        // check

        for ( i = 0; i < 15; i++ ){
                if ( bp->ol_supply_w_id[i] == 0 ) {
                        break;
                }
                if ( bp->ol_i_id[i] == -1 ) {
//                      strcpy( bp->status, "Item number is
not valid" );
                }

                bp->i_name[i][0] = '\0';
                bp->s_quantity[i] = ( rand()%10 )+1;
                bp->brand_generic[i] = ( rand()%26 )+'A';

#ifdef Symfo
                bp->i_price[i] = (long)(( rand()%10000 )+1 );
        // check
#else
                bp->i_price[i] =
(float)((( rand()%10000 )+1 )/100.0); // check
#endif
                bp->ol_amount[i]
```

```
                    = bp->i_price[i] * bp->ol_quantity[i];
                    // check
            bp->total_amount += bp->ol_amount[i];
                        // check
        }
        bp->o_ol_cnt = i;

        return;
}

#endif

//
//

#ifdef DBPRT
void oder_dsp(rte_input_data *in_data,
                    orderstat_trans *bp, int w_id, int
d_flag)
{
   int i;

   if (d_flag == 0){
            fprintf (test_fp, "----- in data area -----\n\n");
            fprintf (test_fp, "w_id = %d ", w_id);
            fprintf (test_fp, "d_id = %s ",in_data-
>D_ID);

            if (in_data->C_ID != 0)
               fprintf (test_fp, "c_id = %s \n",in_data-
>C_ID);
            if (in_data->C_LAST != 0)
               fprintf (test_fp, "c_last = %s \n",in_data-
>C_LAST);

            fprintf (test_fp, "----- trans buf area -----
\n\n");
            fprintf (test_fp, "w_id = %d ", bp->w_id);
            fprintf (test_fp, "d_id = %d ", bp->d_id);
            fprintf (test_fp, "c_id = %d\n", bp->c_id);
            if ( bp->c_last[0] == '\0') {
//             fprintf (test_fp, "byname = %d \n", bp-
>ordin.bylastname);
            } else {
//             fprintf (test_fp, "c_last = %s :byname
= %d\n",bp->ordin.c_last,
//                    bp->ordin.bylastname);
            }
   }
   else {
            fprintf(test_fp, "----- trans buf area (after) ---
--\n\n");
            fprintf(test_fp,"w_id = %d  ", bp->w_id);
      fprintf(test_fp,"d_id = %d  ", bp->d_id);
      fprintf(test_fp,"c_id = %d\n", bp->c_id);
      fprintf(test_fp,"c_first=%s ", bp->c_first);
      fprintf(test_fp,"c_middl=%s ", bp->c_middle);
      fprintf(test_fp,"c_last =%s\n", bp->c_last);

      fprintf(test_fp,"c_balan=%f ", bp->c_balance);
      fprintf(test_fp,"o_id =%d ", bp->o_id);
      fprintf(test_fp,"o_entry_d=%s\n", bp->o_entry_d );
       // check

      if ( bp->o_carrier_id != 0 ) {
         fprintf(test_fp,"o_carrier_id=%d\n", bp-
>o_carrier_id);
      }


      for( i = 0; i < bp->o_ol_cnt; i++ ){

         fprintf(test_fp,"ol_supp=%d ", bp-
>ol_supply_w_id[i]);
         fprintf(test_fp,"ol_i_id=%d ", bp->ol_i_id[i]);
         fprintf(test_fp,"ol_quan=%d ", bp-
>ol_quantity[i]);
         fprintf(test_fp,"ol_amou=%f\n", bp-
>ol_amount[i]);
      }
   }
}

void pay_dsp(rte_input_data *in_data,
                    payment_trans *bp, int w_id, int
d_flag)
{
   int i;

   if (d_flag == 0){
            fprintf (test_fp, "----- in data area -----\n\n");

            fprintf (test_fp, "w_id = %d ",  w_id);
            fprintf (test_fp, "d_id = %s ",in_data-
>D_ID);
            fprintf (test_fp, "c_w_id=%s ",in_data-
>C_W_ID);
            fprintf (test_fp, "c_d_id=%s ",in_data-
>C_D_ID);
            fprintf (test_fp, "h_amount=%s \n",in_data-
>H_AMOUNT);

            if (in_data->C_ID != 0)
                  fprintf (test_fp, "c_id = %s
\n",in_data->C_ID);
            if (in_data->C_LAST != 0)
                  fprintf (test_fp, "c_last = %s
\n",in_data->C_LAST);

            fprintf (test_fp, "----- trans buf area -----
\n\n");
            fprintf (test_fp, "w_id = %d ", bp->w_id);
            fprintf (test_fp, "d_id = %d ", bp->d_id);
            fprintf (test_fp, "c_id = %d ", bp->c_id);
            if ( bp->c_last[0] == '\0') {
//             fprintf (test_fp, "byname = %d \n", bp-
>payin.bylastname);
            } else {
//             fprintf (test_fp, "c_last = %s :byname
= %d\n",bp->payin.c_last,
//                    bp->payin.bylastname);
            }
            fprintf (test_fp, "c_w_id=%d ",bp->c_w_id);
            fprintf (test_fp, "c_d_id=%d ",bp->c_d_id);
            fprintf (test_fp, "h_amount=%f \n",bp-
>h_amount);
      }
   else {
            fprintf (test_fp, "----- trans buf area (after) --
---\n\n");
            fprintf(test_fp,"w_id = %d  ", bp->w_id);
         fprintf(test_fp,"d_id = %d  ", bp->d_id);
            fprintf(test_fp,"c_id = %d\n", bp->c_id);

            fprintf(test_fp,"w_str_1=%s  ",bp-
>w_street_1);
            fprintf(test_fp,"w_str_2=%s\n",bp-
>w_street_2);
            fprintf(test_fp,"d_str_1=%s  ",bp-
>d_street_1);
            fprintf(test_fp,"d_str_2=%s\n",bp-
>d_street_2);
            fprintf(test_fp,"w_city=%s ",  bp->w_city);
            fprintf(test_fp,"w_state=%s\n",bp-
>w_state);
            fprintf(test_fp,"d_city=%s ",  bp->d_city);
            fprintf(test_fp,"d_state=%s\n",bp->d_state);

            fprintf(test_fp,"c_w_id=%d  ", bp->c_w_id);
            fprintf(test_fp,"d_w_id=%d\n", bp->c_d_id);

         fprintf(test_fp,"c_first=%s ", bp->c_first);
            fprintf(test_fp,"c_middl=%s ", bp-
>c_middle);
            fprintf(test_fp,"c_last =%s\n",bp->c_last);

            fprintf(test_fp,"c_str_1=%s  ",bp-
>c_street_1);
            fprintf(test_fp,"c_str_2=%s\n",bp-
>c_street_2);
            fprintf(test_fp,"c_city=%s\n", bp->c_city);
            fprintf(test_fp,"c_credi=%s  ",bp->c_credit);
            fprintf(test_fp,"c_state=%s\n", bp->c_state);

         fprintf(test_fp,"c_balan=%f\n", bp->c_balance);

            i = strlen( bp->c_data );
         fprintf(test_fp,"c_date=%s\n", bp->c_data);
   }
}

void sto_dsp(rte_input_data *in_data,
                    stocklvl_trans *bp, int w_id, int d_id,
int d_flag)
{

   if (d_flag == 0){
            fprintf (test_fp, "----- in data area -----\n\n");

            fprintf(test_fp,"w_id = %d ",  w_id);
            fprintf(test_fp,"d_id = %d  ", d_id);
            fprintf (test_fp, "threshold= %s \n",in_data-
>threshold);

            fprintf (test_fp, "----- trans buf area -----
\n\n");
            fprintf (test_fp, "w_id = %d   ", bp->w_id);
            fprintf (test_fp, "d_id = %d   ", bp->d_id);
            fprintf (test_fp, "threshold= %d \n",bp-
>threshold);
   }
   else{
      fprintf (test_fp, "----- trans buf area (after) -----
\n\n");

      fprintf (test_fp, "w_id = %d   ", bp->w_id);
      fprintf (test_fp, "d_id = %d   ", bp->d_id);
      fprintf (test_fp, "threshold= %d   ",bp->threshold);
      fprintf (test_fp, "low_stock= %d \n",bp-
>low_stock);
   }
}

void new_dsp(rte_input_data *in_data,
```

```
                    neworder_trans *bp, int w_id, int
d_flag, int cnt)
{
   int i, loop;


   if (d_flag == 0){
      fprintf (test_fp, "----- in data area -----\n\n");

            fprintf (test_fp, "w_id = %d ", w_id);
      fprintf (test_fp, "d_id = %s ",in_data->D_ID);
      fprintf (test_fp, "c_id = %s \n", in_data->C_ID);

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

         if( in_data->OL_SUPPLY_W_ID[i] != 0 ){
            fprintf(test_fp,"ol_sup_w_id=%s ",in_data-
>OL_SUPPLY_W_ID[i]);
         }

         if( in_data->OL_I_ID[i] != 0 ){
            fprintf (test_fp, "ol_i_id=%s ", in_data-
>OL_I_ID[i]);
         }

         if( in_data->OL_QUANTITY[i] != 0 ){
            fprintf (test_fp, "ol_quan=%s\n", in_data-
>OL_QUANTITY[i]);
         }
      }

      fprintf (test_fp, "----- trans buf area -----\n\n");
      fprintf (test_fp, "w_id = %d ", bp->w_id);
      fprintf (test_fp, "d_id = %d ", bp->d_id);
      fprintf (test_fp, "c_id = %d\n",bp->c_id);

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

         fprintf (test_fp, "ol_sup_w_id=%d ",  bp-
>ol_supply_w_id[i]);
         fprintf (test_fp, "ol_i_id=%d ",  bp->ol_i_id[i]);
         fprintf (test_fp, "ol_quan=%d\n", bp-
>ol_quantity[i]);
      }
   }
   else{
      fprintf (test_fp, "----- trans buf area (after) -----
\n\n");

      fprintf (test_fp, "c_last=%s  ", bp->c_last);
      fprintf (test_fp, "c_credit=%s\n", bp->c_credit);
      fprintf (test_fp, "o_id=%d  ", bp->o_id);

      fprintf (test_fp, "o_entry_d=%s\n",bp->o_entry_d);
       // check
      fprintf (test_fp, "c_discnt=%f\n", bp->c_discount *
100.0);

      fprintf (test_fp, "o_ol_cnt=%d  ", bp->o_ol_cnt);

      fprintf (test_fp, "w_tax=%f  ", bp->w_tax * 100.0);
      fprintf (test_fp, "d_tax=%f\n", bp->d_tax * 100.0);

      loop = bp->o_ol_cnt;
      for ( i = 0; i < loop; i++ ) {

                  fprintf(test_fp," - - - - - -
\no_sup_w_id=%d ",
                              bp->ol_supply_w_id[i]);
         fprintf(test_fp,"o_i_id=%d  ",bp->ol_i_id[i]);
         fprintf(test_fp,"i_name=%s\n",&bp-
>i_name[i][0]);
         fprintf(test_fp,"o_quant=%d ",bp->ol_quantity[i]);
         fprintf(test_fp,"s_quant=%d ",bp->s_quantity[i]);
         fprintf(test_fp,"brand=%c ",  bp-
>brand_generic[i]);
         fprintf(test_fp,"i_price=%f ",bp->i_price[i]);
        // check
         fprintf(test_fp,"ol_amnt=%f\n",bp-
>ol_amount[i]);     // check
      }
//    fprintf (test_fp, "status=%s\n", bp->status);
      fprintf (test_fp, "total_a=%f\n", bp->total_amount);
       // check
   }
}

#endif

#ifdef TIMEST
int tsp(int id, char flag, char type){

//    struct  tm times;
   SYSTEMTIME systemTime; // for IIS Version

      GetLocalTime(&systemTime);

//    times.tm_year = (int)systemTime.wYear - 1900;
//    times.tm_mon  = (int)systemTime.wMonth - 1;
//    times.tm_mday = (int)systemTime.wDay;
//    times.tm_hour = (int)systemTime.wHour;
//    times.tm_min  = (int)systemTime.wMinute;
//    times.tm_sec  = (int)systemTime.wSecond;

      fprintf (TIMES, "ID=%d, FL=%d,
T=%c : %d:%d:%d.%d\n",
            id, flag, type, (int)systemTime.wHour,
            (int)systemTime.wMinute,
            (int)systemTime.wSecond,
            (long)systemTime.wMilliseconds);

      fflush (TIMES);
      return 0; }

#endif
```

File: ex_tpcc_info.h

```
/*=======================================
==================================+
  FILENAME: ex_tpcc_info.h
  This file is used for the function which refers to the
external variable
 +=======================================
==================================*/

extern  long    olen;
extern  void    *trans_buf;
extern  int     trans_size;
extern  int     svrnum;

/*
extern  int     logincnt;
extern  int     base_cok;
extern  char    NewOrdername[];
extern  char    Paymentname[];
extern  char    OrderStatusname[];
extern  char    Deliveryname[];
extern  char    StockLevelname[];
*/

extern  int     now_cookie;
extern  int     now_w_id;
extern  int     now_d_id;
extern  char    now_ottname[];
extern  char    SOPATH[];

extern  char    s_work[];

#ifdef DBPRT
extern  FILE    *test_fp;
#endif

#include "trans.h"
```

File: fldtbl.h

```
/*      fname       fldid           */
/*      -----   -----           */
#define       FML_TERM        ((FLDID)10193)    /*
number: 2001        type: long */
#define       FML_TRAN((FLDID)10194)    /* number:
2002   type: long */
#define       FML_DATA((FLDID)51155)    /* number:
2003   type: carray */
```

File: menupage.h

```
/* ---------------------------------------------------------------------
-
   menupage.h
   ---------------------------------------------------------------------
- */

/* Transaction select screen data */
#define h_menu "\
<HTML><HEAD><TITLE>TPC-
WINDOW</TITLE></HEAD>\r\n\
<BODY><CENTER>Transaction
Menu<BR></CENTER>\r\n\
<PRE>\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\
\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n</PRE>\
<FORM ACTION=\"%s\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"New
order\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"Order
Status\">\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"Stock
Level\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</FORM></BODY></HTML>\r\n"

/* If client can not connect server then use this format */
#define noconnt "\
<HTML><HEAD><TITLE>ERROR: Can't
Connect</TITLE></HEAD><BODY>\r\n\
<P>Sorry. all %d database connections are currently in
use. <br>\
```

```
Please try again later. terminal number is %d.
</P></BODY></HTML>\r\n"

/* If client can not connect server then use this format */
#define conntmax "\
<HTML><HEAD><TITLE>ERROR: Can't
Connect</TITLE></HEAD><BODY>\r\n\
<P>The terminal number you entered isn't valid.<BR>\
It must be an integer in the range 1 to %d.<P>\r\n\
The terminal number which you specified
is %d.</P></BODY></HTML>\r\n"

/* If client can not connect server then use this format */
#define badterm "\
<HTML><HEAD><TITLE>ERROR: Can't
Connect</TITLE></HEAD><BODY>\r\n\
<P>The terminal number you entered isn't valid.<BR>\
This Client process the terminal of %d to %d.<P>\r\n\
The terminal number which you specified
is %d.</P></BODY></HTML>\r\n"

/* If connect client by WWW browser then use this
format */
/* Terminal verification use olny */
#define loginpage "\
<HTML><HEAD><TITLE>Welcome to TPC-
C : %s</TITLE></HEAD><BODY>\r\n\
<P>Please identify your terminal number for this
session.</P>\r\n\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\" VALUE=\"l\">\r\n\
Your Terminal Number:<INPUT NAME=\"c\"
SIZE=5><BR><HR>\r\n\
<INPUT TYPE=\"submit\">\r\n\
</FORM></BODY></HTML>\r\n"

/* LOGIN ERROR PAGE */
/* If terminal number not find then use this format */
#define loginerr2 "\
<HTML><HEAD><TITLE>ERROR :Welcome to TPC-
C</TITLE></HEAD><BODY>\r\n\
<P>Please identify your terminal number for this
session.</P>\r\n\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\" VALUE=\"l\">\r\n\
Your Terminal Number:<INPUT NAME=\"c\"
SIZE=5><BR><HR>\r\n\
<INPUT TYPE=\"submit\">\r\n\
</FORM></BODY></HTML>\r\n"

/*
#define endpage "\
<HTML><HEAD><TITLE>Welcome to TPC-
C</TITLE></HEAD><BODY>\r\n\
<P>Please identify your Warehouse and District for this
session.</P>\r\n\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\" VALUE=\"l\">\r\n\
Your Warehouse ID: <INPUT NAME=\"W\"
SIZE=4><BR>\r\n\
Your District ID:  <INPUT NAME=\"d\"
SIZE=2><BR><HR>\r\n\
<INPUT TYPE=\"submit\">\r\n\
</FORM></BODY></HTML>\r\n"
*/
```

File: newpage.h

```
/* ---------------------------------------------------------------------
-
  newpage.h
  data of neworder transaction result screen (HTML
form)
  ---------------------------------------------------------------------
- */

/* Header data */
#define h_new1 "\
<HTML><HEAD><TITLE>TPC-
WINDOW</TITLE></HEAD><BODY>\r\n\
<CENTER>New Order<BR></CENTER>\r\n\
<font size=4>\r\n<PRE>"

/* Screen data*/
#define h_new2 "\
Warehouse:       District:                    Date:  -
-      :  :  \r\n\
Customer:        Name:
Credit:     %%Disc:   .  \r\n\
Order Number:          Number of Lines:         W_tax:   .
D_tax:   .  \r\n\r\n\
Supp-W   Item-Id  Item Name              Qty  Stock
B/G  Price    Amount\r\n\
                                          $  .   $   .
\r\n\
                                          $  .   $   .
\r\n\
                                          $  .   $   .
\r\n\
                                          $  .   $   .
\r\n\
                                          $  .   $   .
\r\n\
                                          $  .   $   .
\r\n\
                                          $  .   $   .
\r\n\
                                          $  .   $   .
\r\n\
                                          $  .   $   .
\r\n\
                                          $  .   $   .
\r\n\
                                          $  .   $   .
\r\n\
                                          $  .   $   .
\r\n\
Execution Status:                                Total:
$     .  \r\n"

/* Tailer data */
#define h_new3 "\
</PRE>\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"New
order\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"Order
Status\">\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"Stock
Level\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</FORM></BODY></HTML>"

/* Offset to field which should set data */
int newp[] = {
0xb, 0x1c, 0x3c,
0x5c, 0x69, 0x84, 0x90,
0xa5, 0xc0, 0xd1, 0xe0,
0x138, 0x140, 0x148, 0x163, 0x169, 0x170, 0x175,
0x17e,
0x5f9, 0x62d
};
```

File: odrpage.h

```
/* ---------------------------------------------------------------------
-
  odrpage.h
  data of OrderStatus transaction result screen (HTML
form)
  ---------------------------------------------------------------------
- */

/* Header data */
#define h_order1 "\
<HTML><HEAD><TITLE>TPC-
WINDOW</TITLE></HEAD>\r\n\
<BODY><CENTER>Order-
Status<BR></CENTER>\r\n\
<font size=4>\r\n<PRE>"

/* Screen data*/
#define h_order2 "\
Warehouse:       District:   \r\n\
Customer:        Name:                              \r\n\
Cust-Balance: $      .  \r\n\r\n\
Order-Number:            Entry-Date:   -  -       :  :
Carrier-Number:   \r\n\
Supply-W     Item-Id    Qty     Amount      Delivery-
Date\r\n\
                         $    .         -  -    \r\n\
                         $    .         -  -    \r\n\
                         $    .         -  -    \r\n\
                         $    .         -  -    \r\n\
                         $    .         -  -    \r\n\
                         $    .         -  -    \r\n\
                         $    .         -  -    \r\n\
                         $    .         -  -    \r\n\
                         $    .         -  -    \r\n\
                         $    .         -  -    \r\n\
                         $    .         -  -    \r\n\
                         $    .         -  -    \r\n\
                         $    .         -  -    \r\n\
"

/* Tailer data */
#define h_order3 \
"</PRE>\r\n\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n\
```

```
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"New
order\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"Order
Status\">\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"Stock
Level\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</FORM></BODY></HTML>"

/* Offset to field which should set data */
int orderp[] = {
0x0B, 0x1C,
0x2a, 0x37, 0x49, 0x4c,
0x6c,
0x87,  0x9e,  0xc4,
0x105, 0x110,  0x11b, 0x123, 0x131};   /* offset 0x39 */
```

File: paypage.h

```
/* ----------------------------------------------------------------------
-
   paypage.h
   data of Payment transaction result screen (HTML
form)
   ----------------------------------------------------------------------
- */

/* Header data */
#define h_pay1 "\
<HTML><HEAD><TITLE>TPC-
WINDOW</TITLE></HEAD><BODY>\r\n\
<CENTER>Payment<BR></CENTER><font
size=4>\r\n<PRE>"

/* Screen data*/
#define h_pay2 "\
Date:   - -       : :   \r\n\
\r\n\
Warehouse:                      District:   \r\n\
"

#define h_pay4 "\
                    -                         -   \r\n\
\r\n\
Customer:       Cust-Warehouse:       Cust-District:
\r\n\
Name:                               Since:   - -    \r\n"
/*
                                 Credit:   \r\n\
                                 %%Disc:    .  \r\n\
*/

#define h_pay5 "\
                          -           Phone:        - - -
\r\n\
\r\n\
Amount Paid:          $    .        New Cust-Balance:
$           .  \r\n\
Credit Limit:   $           .  \r\n\
\r\n\
Cust-Data: "
/*
Cust-Data:                                          \r\n\
                                                \r\n\
                                                \r\n\
                                                \r\n"
*/

/* Tailer data */
#define h_pay3 "\
</PRE>\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"New
order\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"Order
Status\">\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"Stock
Level\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</fORM></BODY></HTML>"

/* Offset to field which should set data */
int payp[] = {
0x06,
0x29, 0x51,
0x55, 0x7e,
0x94, 0xbd,
0xd3, 0xe8, 0xeb, 0xf1, 0xfc, 0x111, 0x114, 0x11a,
0x12c, 0x142, 0x157,
0x163, 0x174, 0x177, 0x194,  /* 18 - 21 */
0x1a8, 0x1d9,                /* 22, 23 */
0x1e5, 0x216,                /* 24, 25 */
0x225, 0x23a, 0x23d, 0x256,
0x284, 0x2a4,
0x2c5,
0x2e1,                       /* offset 0x3e */
0x320,
0x35f,
0x39e};
```

File: Resource.h

```
//{{NO_DEPENDENCIES}}
// Microsoft Visual C++ generated include file.
// Used by tpapl.rc
//

#define IDS_SERVER        102

#define _APS_NEXT_RESOURCE_VALUE        103
#define _APS_NEXT_CONTROL_VALUE         103
#define _APS_NEXT_SYMED_VALUE           103
#define _APS_NEXT_COMMAND_VALUE         32768
```

File: StdAfx.cpp

```
// stdafx.cpp :
//      tpapl.pch :
//      stdafx.obj :

#include "stdafx.h"
```

File: StdAfx.h

```
#if !defined(AFX_STDAFX_H__13B36D5C_C1F8_11D1
_8326_00A0C906407A__INCLUDED_)
#define
AFX_STDAFX_H__13B36D5C_C1F8_11D1_8326_00A
0C906407A__INCLUDED_

// stdafx.h :
//

#include <afx.h>
#include <afxwin.h>
#include <afxmt.h>
#include <afxext.h>
#include <afxisapi.h>

//{{AFX_INSERT_LOCATION}}

#endif
// !defined(AFX_STDAFX_H__13B36D5C_C1F8_11D1
_8326_00A0C906407A__INCLUDED)
```

File: stpage.h

```
/* ----------------------------------------------------------------------
-
   stopage.h
   data of Stock Level transaction result screen (HTML
form)
   ----------------------------------------------------------------------
- */

/* Header data */
#define h_stock1 "\
<HTML><HEAD><TITLE>TPC-
WINDOW</TITLE></HEAD><BODY>\
<CENTER>Stock-Level<BR></CENTER>\
<font size=4>\r\n<PRE>"

/* Screen data*/
#define h_stock2 "\
Warehouse:        District:   \r\n\
\r\n\
Stock Level Threshold:   \r\n\
\r\n\
low stock:    \r\n\
\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n\r\n"

/* Tailer data */
#define h_stock3 "\
</PRE>\
<FORM ACTION=\"%s\" METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"New
order\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Payment\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Delivery\">\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"Order
Status\">\
<INPUT TYPE=\"submit\" NAME=\"b\" VALUE=\"Stock
Level\">\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</FORM></BODY></HTML>\n"

/* Offset to field which should set data */
int stockp[] = {
```

```
0x0B, 0x1C,
0x39,
0x4a};
```

File: tpapl.cpp

```
//
//  TPC-C Client Application Program Source
//
//  tpapl Extension
//
#include "stdafx.h"
#include "tpapl.h"

#include "tpcciis.h"
#include "tpcc_info.h"
#include "trans.h"
#include "bench2.h"
#include "dbgprt.h"

// HTML-Page Data
#include "tpcweb.h"
#include "tpcinweb.h"
#include "menupage.h"
#include "newpage.h"
#include "paypage.h"
#include "odrpage.h"
#include "delpage.h"
#include "stpage.h"

#include "ATMI.H"                  // TP-BASE include File

#ifdef USE_FML
#  include "fml.h"
#  include "fldtbl.h"         // Create by mkfldhdr commnd.
#endif

char   *point;
FILE  *envget;
FILE    *errfile;

#include "dmy.h"           // For debug

static TPINIT *tpinf;
static DWORD  TLSIsTpInitedKey;
static int    ThrTpInit();

//
// The date data is converted. (The time data is not
contained.)
// Numeric data is converted into character string data.
//
void convert_time( char *save_p, double time )
{
        struct tm tim;
    time_t    tt = ( time_t )time;

    tim = *( localtime( &tt ) );

    sprintf( save_p, "%02d-%02d-
%04d %02d:%02d:%02d",
             tim.tm_mday, tim.tm_mon+1, tim.tm_year
+ 1900,
       tim.tm_hour, tim.tm_min, tim.tm_sec );
}

//
// The date data is converted. (The time data is
contained.)
// Numeric data is converted into character string data.
//
void convert_date( char *save_p, double time )
{
        struct tm tim;
    time_t    tt = ( time_t )time;

    tim = *( localtime( &tt ) );

        sprintf( save_p, "%02d-%02d-%04d",
                tim.tm_mday, tim.tm_mon+1, tim.tm_year
+ 1900 );
}

/*
   Thread    tpinit( )
           Thread       tpinit( )               TLS

     tpinit( )
     tpinit( )

   Tpinit( ) is executed with each thread.
     The flag in the TLS region is checked,  and whether
corresponding thread has executed
     function tpinit is judged.
     When function tpinit is unexecution,  the flag is not
set.
     When function tpinit is executed,  the flag is set.
*/
static int ThrTpInit()

static int num_tpinits=0;
static int x=1;
static int once=0;
static CRITICAL_SECTION TpCriticalSection;
int    lasterr, iRc, TpRc;
int    retry = 0;
BOOL   Success = FALSE;

        // Whether the key data is set is checked.
        if(!TlsGetValue(TLSIsTpInitedKey)) {

      // If the key data is not set
                FILE *fp;
                fp = fopen("c:\\tuxlog\\tpapl.log", "ab");

        if (!once)

        InitializeCriticalSection(&TpCriticalSection);
                        once=1;

                        fprintf(fp, "- Start ----------------------
\r\n");
                }

#ifdef DBPRT
                fprintf(fp, "* In ThrTpInit Thread %d * \r\n",
GetCurrentThreadId());
#endif
                fclose(fp);

        while ( retry < 10 )

        EnterCriticalSection(&TpCriticalSection);

                        // Execute tpalloc( )
                        if(tpinf == NULL)

                        if ((tpinf = ( TPINIT *)tpalloc("TPINIT",
NULL, sizeof(TPINIT))) == NULL)


        LeaveCriticalSection(&TpCriticalSection);
                                TpRc = tperrno;

                                {
                        FILE *fp;
                                fp =
fopen("c:\\tuxlog\\tpapl.log", "ab");
                                fprintf(fp," > ThrTpInit:%d :
tpalloc of tpinit failed: %d : %s\r\n",
                                        GetCurrentThreadId(),
TpRc, tpstrerror(TpRc));
                        fclose(fp);
                                }
                                retry++;
                                continue;
                        }

                        tpinf-
>flags|=TPMULTICONTEXTS;
                }

                        // Execute tpinit( )
                        iRc = tpinit(tpinf);
                        TpRc = tperrno;

                        // check return code
                        if (iRc < 0)
                                // if tpinit abnormal end

        LeaveCriticalSection(&TpCriticalSection);
                                retry++;
                        lasterr = GetLastError();
                  }
                        else
                                // if tpinit( ) normal end
                                Success = TRUE;

        LeaveCriticalSection(&TpCriticalSection);
                        break;
                }
                        Sleep(5); // Relinquish thread
timeslice
                } // retry the tpinit if it failed the first time

                // if tpinit( ) abnormal end
                if ( Success == FALSE )
                {
                char ebuf[128];
                sprintf(ebuf,
                    "False : ThrTpInit %d : Cannot tpinit
after %d tries iRc = %d LastErr = %d \r\n",
                        GetCurrentThreadId(), 1, iRc, lasterr);

                FILE *fp;
                        fp = fopen("c:\\tuxlog\\tpapl.log",
"ab");
                   fprintf(fp,"%s\n", ebuf);
                fclose(fp);

                }
                return -1;
                }
```

```
            // if tpinit( ) normal end
        if ( Success == TRUE )

                    if ( retry > 0 )
                char ebuf[128];
                sprintf(ebuf,
                        "Success : ThrTpInit %d: Cannot
tpinit after %d tries iRc = %d LastErr = %d\r\n",
                        GetCurrentThreadId(), 10, iRc,
lasterr);
                sprintf(ebuf,
                        "Success : ThrTpInit Thread %d
Success retry count %d with LastErr = %d *\r\n",
                        GetCurrentThreadId(), retry,
lasterr);

                    FILE *fp;
                        fp = fopen("c:\\tuxlog\\tpapl.log",
"ab");
                    fprintf(fp,"%s\n", ebuf);
                fclose(fp);
                    }

                    if
( ( iRc=TlsSetValue(TLSIsTpInitedKey,&x)) == 0) {  //?
                    {
                            FILE *fp;
                            fp = fopen("c:\\tuxlog\\tpapl.log",
"ab");

                            fprintf(fp, " > ThrTpInit %d :
TlsSetValue Failed iRc: %d \r\n",
                                    GetCurrentThreadId(),
iRc);
                            fclose(fp);
                    }
                    }
            }
        }
        else
            // If the key data is set
#ifdef DBORT
            FILE *fp;
            fp = fopen("c:\\tuxlog\\tpapl.log", "ab");
            fprintf(fp, "ThrTpInit Thread %d already
tpinited * (%x) \r\n",
                    GetCurrentThreadId(), tpinf->flags);
            fclose(fp);
#endif
        }

        return 0;
}

/*
        check HTML form

*/

int checkHTMLform( char *str, char *buffer)
{
        int length;
        int cnt1;
        int cnt2 = 0;
        int newlength = 0;

        length = strlen( str );

        for (cnt1 = 0; cnt1 < length; cnt1++){

                if ( *(str + cnt1) == '&' ){
                        *(buffer + cnt2) = '&'; cnt2++;
                        *(buffer + cnt2) = 'a'; cnt2++;
                        *(buffer + cnt2) = 'm'; cnt2++;
                        *(buffer + cnt2) = 'p'; cnt2++;
                        *(buffer + cnt2) = ';'; cnt2++;
                }
                else if ( *(str + cnt1) == '<' ) {
                        *(buffer + cnt2) = '&'; cnt2++;
                        *(buffer + cnt2) = 'l'; cnt2++;
                        *(buffer + cnt2) = 't'; cnt2++;
                        *(buffer + cnt2) = ';'; cnt2++;
                }
                else if ( *(str + cnt1) == '>' ) {
                        *(buffer + cnt2) = '&'; cnt2++;
                        *(buffer + cnt2) = 'g'; cnt2++;
                        *(buffer + cnt2) = 't'; cnt2++;
                        *(buffer + cnt2) = ';'; cnt2++;
                }
                else if ( *(str + cnt1) == '"' ) {
                        *(buffer + cnt2) = '&'; cnt2++;
                        *(buffer + cnt2) = 'q'; cnt2++;
                        *(buffer + cnt2) = 'u'; cnt2++;
                        *(buffer + cnt2) = 'a'; cnt2++;
                        *(buffer + cnt2) = 't'; cnt2++;
                        *(buffer + cnt2) = ';'; cnt2++;
                }
                else {
                        *(buffer + cnt2) =  *(str + cnt1);
cnt2++;
                }
        }

        *(buffer + cnt2) = 0;
        return ( strlen (buffer) );
}


/*
   alp2str : Outputs a string into the memory space
supplied.

   field = the destination field
   field_size = number of characters to output
   string = alpha string to be displayed
*/

void alp2str(char *str, int len, char *alp)
{
    int cnt;

    cnt = strlen (alp);
    strncpy (str, alp, len);    /* copy to destination area */

    /* len                          */
    /* If not coming up to the specified length then set
the blank. */
    if ( len - cnt > 0 )
            memset ( &str[cnt], ' ', len - cnt);
}

/*
   int2str : Converts an integer value to a string of a
specified length and
   outputs the string to the memory buffer supplied.
   field = the destination field
   field_size = number of characters to output
   value = integer to be displayed
*/

void int2str(char *str, int len, int num)
{
    int cnt;

    for (cnt = len - 1; cnt >= 0 ; cnt--){

                str[cnt] = (num % 10) + '0';
                num /= 10;
    }

    for (cnt = 0; cnt < len-1; cnt++){

                if (str[cnt] == '0')
                    str[cnt] = ' ';
                else
                        return;
    }
}

/*
   int3str : Converts an integer value to a string of a
specified length and
   outputs the string to the memory buffer supplied.

   field = the destination field
   field_size = number of characters to output
   value = integer to be displayed
*/

void int3str(char *str, int len, int num)
{
    int cnt;

    for (cnt = len - 1; cnt >= 0 ; cnt--){

                str[cnt] = (num % 10) + '0';
                num /= 10;
    }
}

/*
   date2str : Outputs a date in the supplied buffer in the
following format:
   DD-MM-YYYY

   field = the destination field
   date = date to be converted and displayed
*/
void date2str(char *str, char *time)
{
    int  year, month, day;

#ifdef DBPRT
    fprintf (test_fp, "date2: %s\n", time);
#endif
    sscanf( time, "%d-%d-%d", &day, &month, &year );

    int3str (str, 2, day);
    str[2] = '-';
    int3str (&str[3], 2, month);
    str[5] = '-';
    int3str (&str[6], 4, year);
```

```
}


/*
   time2str:
   Outputs a date and time in the supplied buffer in the
following format:
   DD-MM-YYYY hh:mm:ss

   field = the destination field
   date = date and time to be converted and displayed
*/
void time2str (char *str, char *time)
{
    int  year, month, day, hour, min, sec;

#ifdef DBPRT
    fprintf (test_fp, "time2: %s\n", time);
#endif
    sscanf( time, "%d-%d-%d %d:%d:%d",
          &day, &month, &year, &hour, &min, &sec );

    int3str (str, 2, day);
    str[2] = '-';

    int3str (&str[3], 2, month);
    str[5] = '-';

    int3str (&str[6], 4, year);
    str[10] = ' ';

    int3str (&str[11], 2, hour);
    str[13] = ':';

    int3str (&str[14], 2, min);
    str[16] = ':';

    int3str (&str[17], 2, sec);
}


/*
   dec2str:
   Converts a double precision floating point value to a
string of
   a specified length and outputs the string to the
memory buffer supplied.
   This routine assumes the following restrictions apply:
   Precision is fixed at 2 places to the right of the
decimal point.
   No string length will be less than 4.

   field = the destination field
   field_size = number of characters to output
   value = floating point number to be displayed
*/

void dec2str(char *str, int len, double num)
{
    int dec, sign, i, cnt;
    char *string;

    string = ecvt(num, len-1, &dec, &sign);

    /* dec =              sign =    0,  1,string=
*/

    if ( dec > 0 ) {
                /* if the integer part is not zero ..
Exsample :mum data is 1234.56 */
                cnt = (len - 3) - dec;

        /*                              : "0012" -> "  12" */
        /* If the high-order digit is zero , zero is changed at
the blank */
                for (i = 0; i < cnt; i++){
                    /* pad with blank in the high part of the
number */
                    str[i] = ' ';
                }

                /* The high-order digit set to the output
area:                   */
                for (; i < (len - 3);
                    str[i] = *(string++);
                }
    }
    else
                /* If the integer part is zero ... Exsample:
num data is 0.12 */
                cnt = len - 4;

                for (i = 0; i < cnt; i++){
                        /* pad with blank in the high part of
the number */
                        str[i] = ' ';
                }
                str[i++] = '0';
        }

    str[i++] = '.';

    for (; dec < 0 && i < len; dec++,i++){
                /* pad with 0's in the high part of the
fraction */
                str[i] = '0';
    }

    for (; i < len; i++){
                /* copy the decimal portion (2 places) */
                str[i] = *(string++);
    }
}

/*
   sigdec2str:
   Converts a double precision floating point value to a
string of
   a specified length and outputs the string to the
supplied buffer.
   If the value is negative, the first character will be a
minus sign (-).

   field = the destination field
   field_size = number of characters to output
   value = floating point number to be displayed
*/

void sigdec2str(char *str, int len, double num)
{
    if (num >= 0.0) {
                str[0] = ' ';
                dec2str (&str[1], len - 1, num);
    } else {
                str[0] = '-';
                dec2str (&str[1], len - 1, -num);
    }
}

/*
   zip2str:
   Outputs a zipcode in the supplied buffer in the
following format:
   XXXXX-XXXX

   str = the destination field
   zip = the zipcode to be output
 */

void zip2str (char *str, char *zip)
{
    alp2str (str, 5, zip);
    str[5] = '-';
    alp2str (&str[6], 4, &zip[5]);
}


/*
   phone2str:
   Outputs a phone number in the supplied buffer in the
following format:
   XXXXXX-XXX-XXX-XXXX

   str   = the destination field
   phone = the phone number to be output
*/
void phone2str(char *str, char *phone)
{
    alp2str (str, 6, phone);
    str[6] = '-';

    alp2str (&str[7], 3, &phone[6]);
    str[10] = '-';

    alp2str (&str[11], 3, &phone[9]);
    str[14] = '-';

    alp2str (&str[15], 4, &phone[12]);
}

#define  numcheck(num) ( 0x30 <= num && num <=
0x39 ) /* 0 - 9 */
#define  alpcheck(num) ( 0x41 <= num && num <=
0x5a ) /* A - Z */

/*
   str2int :
   takes a string, makes sure it's not too long, and
ensures that it
   represents an integer.
   If it does, the corresponding int value is returned.

   -3: there is not string data.
   -2: find not character data.
   -1: string data is too many long
*/

int str2int(char *str, int field_len) {
    int x;

    if(str == 0 || !(x = strlen(str))) return -3;
    if(x > field_len){
                if (strchr (str, '%') != 0 )          /* 98.8.3 :
*/
```

```
                return -2;
        else
                return -1;
    }
    else{
        for( ; x ; x--){
            if (!numcheck(str[x-1]))
                return -2;
                }
       }
    }
    return atoi(str);
}

/*
   str2short :
   takes a string, makes sure it's not too long, and
ensures that it
   represents an integer.
   If it does, the corresponding short value is returned.

   -3: there is not string data.
   -2: find not character data.
   -1: string data is too many long
*/

short str2short(char *str, int field_len) {
    int x;

    if(str == 0 || !(x = strlen(str))) return -3;
    if(x > field_len){
        if (strchr (str, '%') != 0 )          /* 98.8.3 :
*/
                return -2;
        else
                return -1;
       }
       else {
        for( ; x; x--){
            if (!numcheck(str[x-1]))
                return -2;
       }
    }
    x = atoi(str);
    return (short)x;
}

/*
   str2str :
   makes sure the string exists and isn't too long.

   -1: string data is too many long
   -2: find not figure data.
    0: there is not string data.
    1: normal end
*/
int str2str(char *str, int field_len) {
    int x;

       if (str == 0 || !(x = strlen (str))) return 0;

    if(x > field_len ) {
        if ( strchr (str, '%') != 0)          /* 98.8.3 :
*/
                return -2;
        else
                return -1;
       }
    else {
        for( ; x ; x--){
            if (!alpcheck(str[x-1]))
                return -2;
       }
    }
    return 1;
}


/*
   str2dbl :
   takes a string, makes sure it's not too long, and
makes sure that it
   represents a floating point number.
   If so, delete the decimal point.
   As a result, the value is increased hundredfold.
   this function is returned integer value.

   !! This function use Payment transaction only.

   -3: there is not string data.
   -2: find not character data.
   -1: string data is too many long

 */
int str2dbl(char *str, int field_len) {
    int   x, len, cnt;
    char  NUM[7];

    char  pointf = 0;
    int   fcnt = 2;             /* */


    if(str == 0 || !(x = strlen(str))) return -3;
    len = x;

    if(x > field_len){
        if (strchr (str, '%' ) != 0)          /* 98.8.3 :
*/
                return -2;
        else
                return -1;
       }
       else{
        /* check string data */
        for(;x;x--) {
            if(numcheck(str[x-1]));
            else if((str[x-1] == '.') && ((len - x) < 3));
            else if((str[x-1] == '-') && (x == 1));
            else if((str[x-1] == '+') && (x == 1));
            else return -2;
        }
    }

    /* delete the decimal point. As a result,do
hundredfold the value.*/
    for (cnt = 0, x = 0; x < len; x++){

       if ( str[x] == '.') {
          /* find the decimal point. set point flag.*/
          pointf = 1;
       } else
          /* set character to work buffer.*/
          NUM[cnt] = str[x]; cnt++;

          /* The figure below the decimal point was
detected */
          if ( pointf == 1 ) {fcnt--;}
       }
    }

    if ( pointf == 1 && fcnt > 0 ){
        /*There was no figure below the decimal
point or only one digit was
          found.:                                    */
        for ( ; fcnt > 0 ; fcnt--) {
            NUM[cnt++] = '0';
        }
    }
    else if ( pointf == 0 ) {
        /* There is no decimal point.:
*/
        NUM[cnt++] = '0'; NUM[cnt++] = '0';
    }

    NUM[cnt] = 0;

    return (atoi(NUM));
}

/*
   struct_init :
   init_ptrs sets everything in the annoyingly long
raw_form_data structure
   to zero.
*/

void strct_init (rte_input_data *in_data) {

    int  cnt = 0;

    in_data->button = 0;
    in_data->cookie = 0;
    in_data->form = 0;
    in_data->O_CARRIER_ID = 0;
    in_data->threshold = 0;
    in_data->D_ID = 0;
    in_data->C_ID = 0;
    in_data->C_W_ID = 0;
    in_data->C_D_ID = 0;
    in_data->C_LAST = 0;
    in_data->H_AMOUNT = 0;

    for (cnt = 0; cnt < 15; cnt++)
        in_data->OL_SUPPLY_W_ID[cnt] = 0;

    for (cnt = 0; cnt < 15; cnt++)
       in_data->OL_I_ID[cnt] = 0;

    for (cnt = 0; cnt < 15; cnt++)
       in_data->OL_QUANTITY[cnt] = 0;
}

/*
   para_split :

       (QueryString)

       :                                NULL
                NULL

   Split divides up a string based on the first instance of
a specified
   delimiter ('sp').  The first instance of 'sp' is converted
to a NULL
```

```
   and the address of the first character of the second
half is returned.
   Thus the user has the first half (which he passed in
and still has) and
   the second half (which was returned) with a NULL
between them.  Yay.
   (Yes, strtok does this, sort of, but I can't nest strtok
calls.)
*/

char *para_split(char *para, char delimita) {

    char *point;

    /* The address of the delimitation character is
calculated */
    /*                          */
    if ((point = strchr (para, delimita)) == NULL)
        return (char *)0;

    /* The delimitation character is replaced with NULL*/
    *point = '\0';              /*                      */

    /* The first position of the analyzed variable is
returned.*/
    return (point + 1);         /*                      */
}

/*
   anly_para :
   QueryString



        :

   Gets the query string and finds every variable=value
pair contained
   within it.  For every pair, it runs the variable name
through a really
   big compound switch statement that matches for
specific variables we
   want to catch.  When we find a known variable name,
we stick a pointer
   to its corresponding value into the appropriate
member of 'ptrs.'

   query - a 1024 byte buffer that contains the query
string.
   ptrs - a raw_form_data structure to hold pointers.
*/

int anly_para (char *para, rte_input_data *in_data) {
    char *val, *rest;
    int  num = 0;

    if(!para) return 0;

    while(para) {

                rest = para_split(para, '&'); /* next parameta
point */
                val  = para_split(para, '='); /* now value
point */

                switch(para[0]) {
                case 'c':
                    in_data->cookie = val;          break;

                case 'b':
                    in_data->button = val;          break;

                case 'f':
                    in_data->form = val;    break;

                case 't':
                    in_data->threshold = val;   break;

                case 'D':
                    in_data->D_ID = val;    break;

                case 'H':
                    in_data->H_AMOUNT = val;        break;

                case 'C':
                        switch(para[1]) {
                        case 'I':
                                in_data->C_ID = val;    break;

                        case 'W':
                                in_data->C_W_ID = val;break;

                        case 'L':
                                in_data->C_LAST = val;break;

                        case 'D':
                                in_data->C_D_ID = val; break;
                        }
                        break;

                case 'O':
                        switch(para[1]) {
                        case 'C':
                                in_data->O_CARRIER_ID =
val;           break;

                        case 'S':
                                switch(para[2]) {
                                case '0':
                                        if (para[3] >= 0x31 &&
para[3] <= 0x39 ){
                                                num = (int)(para[3]
- 0x30);
                                                if (strlen(val) != 0 )
                                                        in_data-
>OL_SUPPLY_W_ID[num - 1] = val;
                                        }
                                        break;

                                case '1':
                                        if (para[3] >= 0x30 &&
para[3] <= 0x35 ){
                                                num = (int)(para[3]
- 0x30) + 10;
                                                if (strlen(val) != 0 )
                                                        in_data-
>OL_SUPPLY_W_ID[num - 1] = val;
                                        }
                                        break;
                                }
                                break;

                        case 'I':
                                switch(para[2]) {
                                case '0':
                                        if (para[3] >= 0x31 &&
para[3] <= 0x39 ){
                                                num = (int)(para[3]
- 0x30);
                                                if (strlen(val) != 0 )
                                                        in_data-
>OL_I_ID[num - 1] = val;
                                        }
                                        break;

                                case '1':
                                        if (para[3] >= 0x30 &&
para[3] <= 0x35 ){
                                                num = (int)(para[3]
- 0x30) + 10;
                                                if (strlen(val) != 0 )
                                                        in_data-
>OL_I_ID[num - 1] = val;
                                        }
                                        break;
                                }
                                break;

                        case 'Q':
                                switch(para[2]) {
                                case '0':
                                        if (para[3] >= 0x31 &&
para[3] <= 0x39 ){
                                                num = (int)(para[3]
- 0x30);
                                                if (strlen(val) != 0 )
                                                        in_data-
>OL_QUANTITY[num - 1] = val;
                                        }
                                        break;

                                case '1':
                                        if (para[3] >= 0x30 &&
para[3] <= 0x35 ){
                                                num = (int)(para[3]
- 0x30) + 10;
                                                if (strlen(val) != 0 )
                                                        in_data-
>OL_QUANTITY[num - 1] = val;
                                        }
                                        break;
                                }
                                break;
                        }
                        break;
                }

                para = rest;
    }

    if (in_data->cookie != 0)
        return(atoi (in_data->cookie));
    else
        return(0);
}

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

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

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

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

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

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

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

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

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

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

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

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

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

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

"Supply Warehouse ID",
                        /* 14 */

"Item ID",
                                /* 15 */

"Quantity",
                                /* 16 */

"Your entry was outside the range.",
                /* 17 */
"You didn't entry anything for the field.",
                /* 18 */
"Your entry contained too many characters.",
                /* 19 */
"The input data is wrong data type, must be numeric.",
        /* 20 */
"It must be an integer in the range 1 to %d.",
                /* 21 */
"The input data is wrong data type, must be english
capital letter.",      /* 22 */
};

/*
  set_errpage:

  RTE

  a generic error page generator. If the user does
anything screwy,
  s/he gets here.  The function generates an error page
based on the
  two errlvl arguments and returns it for the user..

  When err_no is 13 or more, Order Line Data is
Abnormal.
  ( err_no is the error data line number )

  98.8.3 :
 */

int set_errpage (char *buf, int user, int err_no, int err_inf,
int sub_inf, int sub_inf2) {
   char errmsg[1024];
   int  nchar;
      int      length;

   if(err_no >= 13) {              /* OrderLIne
Data(Neworder) is Abnormal */
        switch(err_inf) {
        case -5:                /* S_W_ID data is
abnormal */
                sprintf(errmsg,errstrings[13],err_no-
12,errstrings[14],errstrings[20]);
                sub_inf2 = maxwh;
                break;
        case -8:                /* S_W_ID data is
uninput */
                sprintf(errmsg,errstrings[13],err_no-
12,errstrings[14],errstrings[18]);
                sub_inf2 = maxwh;
                break;
        case -15:               /* S_W_ID data is
outside range */
                sprintf(errmsg,errstrings[13],err_no-
12,errstrings[14],errstrings[17]);
                sub_inf2 = maxwh;
                break;

        case -1:                /* I_ID data is
uninput */
                sprintf(errmsg,errstrings[13],err_no-
12,errstrings[15],errstrings[18]);
                sub_inf2 = 100000;
                break;
        case -6:                /* I_ID data is
abnormal */
                sprintf(errmsg,errstrings[13],err_no-
12,errstrings[15],errstrings[20]);
                sub_inf2 = 100000;
                break;
        case -16:               /* I_ID data is
outside range */
                sprintf(errmsg,errstrings[13],err_no-
12,errstrings[15],errstrings[17]);
                sub_inf2 = 100000;
                break;

        case -7:                /* Quantity data is
abnormal */
                sprintf(errmsg,errstrings[13],err_no-
12,errstrings[16],errstrings[20]);
                sub_inf2 = 10;
                break;
        case -2:                /* Quantity data is
uninput */
                sprintf(errmsg,errstrings[13],err_no-
12,errstrings[16],errstrings[18]);
                sub_inf2 = 10;
                break;
        case -17:               /* Quantity data is
outside range */
                sprintf(errmsg,errstrings[13],err_no-
12,errstrings[16],errstrings[17]);
                sub_inf2 = 10;
                break;

        default:
                break;
        }

        length = strlen(errmsg);
        sprintf(&errmsg[length], errstrings[21],
sub_inf2);
        sprintf(buf, errhtml, errmsg, SOPATH,
user);

        }
        else if ( err_no == 4 || err_no == 9 || err_no
== 12 ) {

                switch(err_inf) {
                case -3: /* There is not Input data */
                        sprintf(errmsg,
errstrings[err_no], errstrings[18], sub_inf2);
                        break;

                case -1: /* too many characters */
                        sprintf(errmsg,
errstrings[err_no], errstrings[19], sub_inf2);
                        break;

                case -2: /* Not all digits */
                   sprintf(errmsg, errstrings[err_no],
errstrings[20], sub_inf2 );
                        break;

                case -4: /* nothing sub message */
                        sprintf(errmsg,
errstrings[err_no], " ", sub_inf2);
                        break;

                default: /* Other error */
                        sprintf(errmsg,
errstrings[err_no], errstrings[17], sub_inf2);
```

```
                        break;
                }

                sprintf(buf, errhtml, errmsg, SOPATH,
user);
                printf("%s", buf);

        }
        else{
                switch(err_inf) {
                case -3: /* There is not Input data */
                        sprintf(errmsg,
errstrings[err_no], errstrings[18]);
                        break;

                case -1: /* too many characters */
                        sprintf(errmsg,
errstrings[err_no], errstrings[19]);
                        break;

                case -2: /* Not all digits */
                        if (err_no == 7)
                                sprintf(errmsg,
errstrings[err_no], errstrings[22]);
                        else
                                sprintf(errmsg,
errstrings[err_no], errstrings[20]);

                        break;

                case -4: /* nothing sub message */
                        sprintf(errmsg,
errstrings[err_no], " ");
                        break;

                default: /* Other error */
                        sprintf(errmsg,
errstrings[err_no], errstrings[17]);
                        break;
                }

                sprintf(buf, errhtml, errmsg, SOPATH,
user);
                printf("%s", buf);
        }

    DBGR(fprintf (test_fp, "This Transaction is parameter
ERROR\n"));
    return 0;
}

/*
   set_tuxerr :
   this function make error message of the TP-
application program.
*/
int set_tuxerr (char *page, char *err_inf, int cookie) {

#ifndef SCRTEST

    tpterm ( );

#endif

    sprintf(page, tuxerr, err_inf, SOPATH, cookie);

    return 0;
}

/*
   set_oraerr :
   this function make error message of the Oracle
application program.
*/
int set_oraerr (char *page, char *err_inf, int cookie ) {

#ifdef Symfo
    sprintf(page, symfoerr, err_inf, SOPATH, cookie );
#else
    sprintf(page, oraerr, err_inf, SOPATH, cookie );
#endif

    return 0;
}


/*
   set_symfoerr :
   this function make error message of the Oracle
application program.
*/
void set_symfoerr (char *page, int errorpos , int
sqlstate , int cookie )
{
        char *sqlfunc[4] ={"Failure on insert of a new
record",
                        "Failure on select of an existing record",
                        "Failure on update of an existing record",
                        "Failure to delete an existing record"};
        char buf[80];
        int  pos;

    if ( errorpos == 0 ){
                sprintf( buf,"SQLERROR occured ...
( SQLSTATE : %05d )",sqlstate );
    }
    else{
        pos = errorpos / 100 ;
                sprintf( buf,"%s ...
( SQLSTATE : %05d )",sqlfunc[pos-1],sqlstate ) ;
    }

    sprintf(page, symfoerr, buf, SOPATH, cookie );
}


#ifndef USE_FML
//
void term_id ( int cookie ){

    int transaction_id;
    int num;
        int loop;

        //
    for (loop = 0; loop < 5; loop++){

                num = Term_Base + TRN_ID[loop];
                transaction_id = 1;

                while ( num <= cookie ) {
                        num += TRN_ID[loop];
                        transaction_id++;
                }

            srv->m_tcctxt[cookie - Term_Base].trn_id[loop]
= (char)transaction_id;
    }

//    DBGP(fprintf(test_fp," svn=%d ott-cnt=%d,
num=%d)\n", svnum, loop, num));

    return;
}
#endif


/* ----------------------------------------------------------------------
---
   The function number of the TP application program
which requests processing
   is acquired.  ( Get TPCCxx name in tpccsvr.ott )
   ----------------------------------------------------------------------
-- */
int getsvrnam ( int cookie ){

    int svnum = 1;
    int num;
        int loop = 0;

    num = 0;

    if ( maxconnect < cookie || cookie == 0){
        DBGP(fprintf (test_fp, "Term NO(%d) is not
support!\n", cookie));
                return(-1);
    }
/*  COMMENT OUT : 98.01.13
=======================================
=
    for (cnt = 1; cnt < tpc_area->clent_num; cnt++){
        if ( cookie < tpc_area->clent[cnt][0]){
            break;
        }
    }

    cnt--;
    num = tpc_area->clent[cnt][0];
    DBGP(fprintf (test_fp,"cookie=%d (client=%d,
max=%d, num=%d ->",\
                cookie, cnt, tpc_area->clent[cnt][0], num));

    for (loop = 0; loop < tpc_area->clent[cnt][1]; loop++){

        num += tpc_area->ott[cnt][loop];

        if (num > cookie){
            break;
        }
        else{
            svnum++;
        }
    }
=======================================
==============================*/
    DBGP(fprintf(test_fp," svn=%d ott-cnt=%d,
num=%d)\n", svnum, loop, num));

    return (svnum);
}

/*
   fast_menu:
```

```
   This function reads a user's responses to the login
form, sets
   up the user context, and returns the menu page.
*/

int fast_menu ( char *s_buf, rte_input_data *in_data, int
cookie){

    int  w_id, d_id, user_id;
    char flag=0;

        // make w_id, d_id
    w_id = (cookie - 1)/10 + 1;
    d_id = (cookie - 1)%10 + 1;

#ifndef USE_FML
        //
        term_id ( cookie );
#endif

        // execute tpalloc ... : set w_id, d_id, and
trans_buffer pointer
        if ((user_id = srv->Terminit (w_id, d_id, cookie))
== -1){

                // tpalloc terminated abnormaly
                sprintf(s_buf, tuxierr, "tpalloc", SOPATH);
                return -1;
        }

    sprintf(s_buf, h_menu, SOPATH, cookie);
    return 0;
}

/* ----------------------------------------------------------------
    The w_id, d_id are acquired. and call the function
getsvnam
   ----------------------------------------------------------------
*/
int idget ( char *s_buf, int cookie ){

    int  w_id, d_id, user_id, sarvice_name;

    /* make w_id, d_id */
    w_id = (cookie - 1)/10 + 1;
    d_id = (cookie - 1)%10 + 1;

    if ((sarvice_name = getsvrnam ( cookie )) < 0 ){

                /*The terminal number exceeded the
maximum value */
                sprintf(s_buf, noconnt, maxconnect,
cookie);
                return -1;
    }

    sprintf(s_buf, h_menu, SOPATH, cookie);
    return 0;
}


#define  SUPPLY_NG     0x01
#define  I_ID_NG       0x02
#define  QUANTITY_NG   0x04

/* ----------------------------------------------------------------
   chk_NOdata :

   VerifyNewOrderLine verifies that a user's inputs for a
line in
   the New Order form are okay.
      return -5  : w_id abnormal value : Not Number
      return -6  : i_id abnormal value : Not Number
      return -7  : ol_quantity abnormal value : Not
Number

   98.8.3 :                    (-15, -16, -17    : outside
range )

   ----------------------------------------------------------------
- */
int chk_NOdata (neworder_trans *bp, int cnt,
rte_input_data *in_data, int svcnt)
{

    char flag = 0;

    if( in_data->OL_SUPPLY_W_ID[cnt] == 0 &&
in_data->OL_I_ID[cnt] == 0 &&
        in_data->OL_QUANTITY[cnt] == 0 ){

         /* Find last order line : 1                */
    /* comment out : 98.08.25
        bp->ol_i_id[cnt] = 0;
        bp->ol_quantity[cnt] = 0;
        bp->ol_supply_w_id[cnt] = 0;
            return 0;

         */
            return 16;        /* change return code */
    }

    if( in_data->OL_SUPPLY_W_ID[cnt] != 0 ){

        if((bp->ol_supply_w_id[svcnt] =
                    str2int (in_data-
>OL_SUPPLY_W_ID[cnt], 4)) < 1 )  /* 99.6.8 no_range
*/
                return -5;                /* w_id abnormal */

/*
        if((bp->ol_supply_w_id[svcnt] =
            str2int (in_data->OL_SUPPLY_W_ID[cnt], 4)) <
1 ||
                    bp->ol_supply_w_id[svcnt] > maxwh )
{

            if (bp->ol_supply_w_id[svcnt] < 0)
                return -5;                /* w_id abnormal *
              else
                return -15;                    /* outside
range *
              }
*/  /* 99.6.8 */

        }
    else {
            flag |= SUPPLY_NG;
        }

    if( in_data->OL_I_ID[cnt] != 0 ){

              if((bp->ol_i_id[svcnt] =
                    str2int (in_data->OL_I_ID[cnt], 6)) <
0 )           /* 99.6.8 no_range */
                return -6;                /* i_id abnormal
value */

/*        if((bp->ol_i_id[svcnt] =
                    str2int (in_data->OL_I_ID[cnt], 6)) < 0
||
                    bp->ol_i_id[svcnt] > 100000 ) {

          if (bp->ol_i_id[svcnt] < 0)
                return -6;                /* i_id abnormal
value *
              else
               return -16;                /* outside range *
               }
               else if (bp->ol_i_id[svcnt] == 0){
                    /* Convert 0 into -1. if this function
set 0 then the TP application
                      send the return code of abnormal
end : Oracle use only *
#ifdef Oracle
                    bp->ol_i_id[cnt] = -1;
#endif
               }
*/  /* 99.6.8 */
        }
        else{
            flag |= I_ID_NG;
        }

        if( in_data->OL_QUANTITY[cnt] != 0 ){

       if(((bp->ol_quantity[svcnt] =
          str2int (in_data->OL_QUANTITY[cnt], 2)) < 1) ||
                bp->ol_quantity[svcnt] > 10 ){

                    if ( bp->ol_quantity[svcnt] < 0 )
                        return -7;                /*
ol_quantity abnormal value */
                    else
                        return -17;                /*
outside range */
               }
        }
    else{
            flag |= QUANTITY_NG;
        }

    if (flag != 0){

               /* the order lien data is abnormal : there is
a uninput item */
        DBGR(fprintf(test_fp, "neworder ol data check
flag=%d\n",flag));

        if((flag & SUPPLY_NG) != 0 )   return -8;
        if((flag & I_ID_NG) != 0)      return -1;
        if((flag & QUANTITY_NG) != 0)  return -2;
               return 1;
    }
    else{
         /* the order lien data is normal */
            return 1;
    }
}

/* ----------------------------------------------------------------
   setNOdata : This function set the execution result
data of the TP
```

```
            applicatin program.

   OF is an offset value to the next line data.
   cnt is line number
 ------------------------------------------------------------------------
- */

int setNOdata (char *s_work,int OF,int cnt,
                        neworder_trans *bp,rte_input_data
*in_data)
{
#ifdef Symfo

        int2str((s_work + OF + newp[11]), 4, (int)bp-
>ol_supply_w_id[cnt]);

        int2str((s_work + OF + newp[12]), 6, (int)bp-
>ol_i_id[cnt]);

        alp2str((s_work + OF + newp[13]), 24,bp-
>i_name[cnt]);

        int2str((s_work + OF + newp[14]), 2, (int)bp-
>ol_quantity[cnt]);
        int2str((s_work + OF + newp[15]), 3, (int)bp-
>s_quantity[cnt]);
        alp2str((s_work + OF + newp[16]), 1, &bp-
>brand_generic[cnt]);

        dec2str((s_work + OF + newp[17]), 6,
                (double)((double)bp->i_price[cnt] /
(double)100)); // check
        dec2str((s_work + OF + newp[18]), 7,
                (double)((double)bp->ol_amount[cnt] /
(double)100)); // check

        return 0;
#else

    if(!bp->ol_i_id[cnt] ) {
                alp2str ((s_work + OF + newp[11]), 78, " ");
                return -1;
    }
        else {
                int2str((s_work + OF + newp[11]), 4,
(int)bp->ol_supply_w_id[cnt]);

                if (bp->ol_i_id[cnt] == -1 )
                        bp->ol_i_id[cnt] = 0;

                alp2str((s_work + OF + newp[12]), 6,
in_data->OL_I_ID[cnt]);

                alp2str((s_work + OF + newp[13]), 24,bp-
>i_name[cnt]);

                int2str((s_work + OF + newp[14]), 2,
(int)bp->ol_quantity[cnt]);
                int2str((s_work + OF + newp[15]), 3,
(int)bp->s_quantity[cnt]);
                alp2str((s_work + OF + newp[16]), 1, &bp-
>brand_generic[cnt]);

                dec2str((s_work + OF + newp[17]),
6,(double)bp->i_price[cnt]); // check
                dec2str((s_work + OF + newp[18]),
7,(double)bp->ol_amount[cnt]); // check
                return 0;
        }
#endif
}

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

  neworder : this function proccesses the NewOrder
transaction.

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

int neworder (char *s_buf, rte_input_data *in_data, int
cookie)
{
    neworder_trans *bp;
    long olen;
    int  user_id, i;
    int  ol_cnt, cnt, rtn;

        char S_WORK[WORK_S];
        char TPAPL[12];
        char time_data[64];

#ifdef USE_FML
        neworder_trans    tbuf;
        int               w_id;

        bp = &tbuf;
        user_id = cookie - Term_Base;
        memset (bp, 0, sizeof(neworder_trans));
        /* 98.7.29 */
#else
        user_id = cookie - Term_Base;
    bp = ( neworder_trans * )srv-
>m_tcctxt[user_id].trans_b;
#endif

    bp->tx_type = TX_NEWORDER;
        bp->C_R = 0;

#ifdef USE_FML
        sprintf (TPAPL, "TPCC");        // TP-Base
Aplication Name (View)
#else
        sprintf (TPAPL, "TPCC%d", (int)srv-
>m_tcctxt[user_id].trn_id[0]);
#endif

        if ((rtn = ThrTpInit( )) < 0 ){
       sprintf( S_WORK, "Thread init abort NEW (%d) \n",
rtn);
        set_oraerr( s_buf, S_WORK, cookie );
        return (-1);
        }

/* ------------------------------------------------------ check the
Input data */
    bp->w_id = (short)srv->m_tcctxt[user_id].w_id;

        /* 99.6.8 no_range */
//    if((bp->d_id = str2int (in_data->D_ID, 2)) < 1 || bp-
>d_id > 10 )
    if((bp->d_id = str2int (in_data->D_ID, 2)) < 1 )
        return set_errpage(s_buf, cookie, 2, (int)bp->d_id,
0, 0);

    /* 98.8.3 :                    */
//    if((bp->c_id = str2int (in_data->C_ID, 4)) < 1 || bp-
>c_id > 3000 )
    if((bp->c_id = str2int (in_data->C_ID, 4)) < 0 )
                return set_errpage(s_buf, cookie, 6, bp-
>c_id, 0, 0);

        /* 98.8,25 */
    ol_cnt = 0;
    for (cnt = 0; cnt < 15; cnt++){

        if ((rtn = chk_NOdata( bp, cnt, in_data, ol_cnt)) <
0 ){
            return set_errpage(s_buf, cookie, 13 + cnt, rtn,
0, 0);
        }
/*
        else if (rtn == 0 && ol_cnt == 0){
            return set_errpage(s_buf, cookie, 13 + cnt, -8, 0,
0);
        }
        else if (rtn == 0){
            break;
        }
*/
        else if (rtn == 1){
                        ol_cnt++;
        }

//      else // Order Line data is normal: rtn == 1
//          ol_cnt++;
//       }

    }

    /* nothing order line data */
    if ( cnt >= 15 && ol_cnt == 0 )
            return set_errpage(s_buf, cookie, 13, -8, 0, 0);

    /* if ol_cnt < 15 then the last order line set NULL */
    if ( ol_cnt < 15 ){
        bp->ol_i_id[ol_cnt] = 0;
        bp->ol_quantity[ol_cnt] = 0;
        bp->ol_supply_w_id[ol_cnt] = 0;
    }

// |_

    bp->o_ol_cnt = ol_cnt;

        DBGR(new_dsp(in_data, bp, srv-
>m_tcctxt[user_id].w_id, 0, ol_cnt));

/* ---------------------------------------------- Execute NewOrder
transaction */
resend_neworder:

#ifndef SCRTEST

    DBGR(tsp(0));

#ifdef USE_FML
        w_id = bp->w_id;

    rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b,
FML_TERM, 0, (char *)&w_id, 0 );
    rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b,
FML_TRAN, 0, (char *)&bp->tx_type, 0 );
```

```
    rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b,
FML_DATA, 0, (char *)bp,
                ( FLDLEN )sizeof( neworder_trans ) );

    if ( tpcall( TPAPL, ( char * )srv-
>m_tcctxt[user_id].trans_b, 0,
         ( char ** )&srv->m_tcctxt[user_id].trans_b,
                &olen, 0|TPNOTIME ) == -1 ){

        if ( tperrno == TPESVCFAIL ) {
            sprintf( S_WORK, "Oracle failed to process
NewOrder Transaction.\n"
                "tperrno = %d svc = '%s' d_id = %d c_id = %d
lines = %d\n",
                tperrno, TPAPL,bp->d_id, bp->c_id, ol_cnt );

            set_oraerr( s_buf, S_WORK, cookie );
            return (-1);
        }

        sprintf( S_WORK, "tpcall failed in NewOrder:
tperrno = %d\n"
                " svc = '%s' d_id = %d c_id = %d lines
= %d\n",
                tperrno, TPAPL,
                        bp->d_id, bp->c_id, ol_cnt );

        set_tuxerr(s_buf, S_WORK, cookie);
        return (-1);
         }

    DBGR(tsp(1));
         tbuf = *((neworder_trans *)Ffind( (Fbfr *)srv-
>m_tcctxt[user_id].trans_b, FML_DATA, 0, NULL));
         bp = &tbuf;

#else

         if ( tpcall( TPAPL, ( char * )srv-
>m_tcctxt[user_id].trans_b,
                sizeof( neworder_trans ),
                ( char ** )&srv->m_tcctxt[user_id].trans_b,
&olen, 0|TPNOTIME ) == -1 ) {

        if ( tperrno == TPESVCFAIL ) {
            sprintf( S_WORK, "Oracle failed to process
NewOrder Transaction.\n"
                "tperrno = %d svc = '%s' d_id = %d c_id = %d
lines = %d\n",
                tperrno, TPAPL,
                        bp->d_id, bp->c_id, ol_cnt );

            set_oraerr( s_buf, S_WORK, cookie );
            return (-1);
        }

        sprintf( S_WORK, "tpcall failed in NewOrder:
tperrno = %d\n"
                " svc = '%s' d_id = %d c_id = %d lines
= %d\n",
                tperrno, TPAPL,
                        bp->d_id, bp->c_id, ol_cnt );

        set_tuxerr(s_buf, S_WORK, cookie);
        return (-1);
    }
    DBGR(tsp(1));

    bp = ( neworder_trans * )srv-
>m_tcctxt[user_id].trans_b;
#endif

#else
    dummy_neworder( bp );
#endif

    DBGR(new_dsp(in_data, bp, srv-
>m_tcctxt[user_id].w_id, 1, 1));

    sprintf (S_WORK, h_new2);

    int2str ((S_WORK + newp[0]), 4, (int)bp->w_id);
    int2str ((S_WORK + newp[1]), 2, (int)bp->d_id);
    int2str ((S_WORK + newp[3]), 4, bp->c_id);

    alp2str ((S_WORK + newp[4]), 16, bp->c_last);
    alp2str ((S_WORK + newp[5]), 2, bp->c_credit);
    int2str ((S_WORK + newp[7]), 8, (int)bp->o_id);

    if ( bp->C_R == 1 || bp->C_R == 2 ){ // Normal End

        cnt = bp->o_ol_cnt;

#ifdef Symfo
            convert_time (time_data, bp->o_entry_d);
        time2str((S_WORK + newp[2]),time_data);
            dec2str ((S_WORK + newp[6]),5,
                (double)((double)(bp->c_discount) /
(double)100.0) );

        int2str ((S_WORK + newp[8]),2,(int)bp->o_ol_cnt);

        dec2str ((S_WORK + newp[9]),5,
                (double)((double)(bp->w_tax) /
(double)100.0));
        dec2str ((S_WORK + newp[10]),5,
                (double)((double)(bp->d_tax) /
(double)100.0));
#else
        time2str((S_WORK + newp[2]),bp->o_entry_d);
            dec2str ((S_WORK +
newp[6]),5,(double)(bp->c_discount*100.0));
        int2str ((S_WORK + newp[8]),2,(int)bp->o_ol_cnt);
            dec2str ((S_WORK + newp[9]),5,
(double)(bp->w_tax * 100.0));
            dec2str ((S_WORK +
newp[10]),5,(double)(bp->d_tax * 100.0));
#endif

        for ( i = 0; i < cnt; i++ ) {
            setNOdata (S_WORK, 0x50*i, i, bp, in_data);
        }

#ifdef Symfo
            if (bp->C_R == 2)
                alp2str ((S_WORK + newp[19]), 24,
"Item number is not vaild");

            dec2str ((S_WORK + newp[20]), 8,
                (double)((double)(bp->total_amount) /
(double)100.0));
#else
        /* "Item number is not valid" or "" ('\0') */
            // Oracle Web Server use
            alp2str ((S_WORK + newp[19]), 24, bp-
>status);
            dec2str ((S_WORK + newp[20]), 8,
(double)(bp->total_amount));  // check
#endif

    }

#ifndef SCRTEST
    else{

#ifndef Symfo
        if ( bp->newout.terror == IRRECERR ){
            sprintf (S_WORK, "Irrecoverable error in
NewOrder\n" );
                set_tuxerr (s_buf, S_WORK, cookie);
                return (-1);
            }
            else{
          goto resend_neworder;          /* Retry
NewOrder transaction */
            }
#else
            set_symfoerr (s_buf, bp->errorpos, bp-
>sqlstate, cookie) ;
            return (-1);
#endif
    }
#endif

/* ------- The execution result data notified RTE is make
by the HTML form */

    sprintf(s_buf, h_new1);
    strcat (s_buf, S_WORK);

    sprintf(S_WORK, h_new3, SOPATH, cookie);
    strcat (s_buf, S_WORK);

    return (0);
}


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

  payment : this function proccesses the Payment
transaction.

  ----------------------------------------------------------------
-*/
int payment (char *s_buf, rte_input_data *in_data, int
cookie)
{
    payment_trans *bp;
    int    i, user_id, rtn;
    long   olen;

    float  h_amount;                /* For work */

    char   c_id_flag = NG;
        char   S_WORK[WORK_S];
        char   TPAPL[12];
        char   time_data[64];

        char   buffer[128];         /* check HTML form */
        char   buffer2[128];
        char   buffer3[512];
        int        newlength;
```

```
#ifdef USE_FML
    payment_trans   tbuf;
        int             w_id;
        bp = &tbuf;
        user_id = cookie - Term_Base;
        memset (bp, 0, sizeof(payment_trans));
        /* 98.7.29 */
#else
        user_id = cookie - Term_Base;
    bp = ( payment_trans * )srv->m_tcctxt[user_id].trans_b;
#endif

    bp->tx_type = TX_PAYMENT;
        bp->C_R = 0;

#ifdef USE_FML
        sprintf (TPAPL, "TPCC");        // TP-Base Aplication Name (View)
#else
        sprintf (TPAPL, "TPCC%d", (int)srv->m_tcctxt[user_id].trn_id[1] );
#endif

        if ((rtn = ThrTpInit( )) < 0 ){
        sprintf( S_WORK, "Thread init abort PAY (%d) \n", rtn);
        set_oraerr( s_buf, S_WORK, cookie );
        return (-1);
        }

/* ------------------------------------------------------ check the Input data */
    bp->w_id = (short)srv->m_tcctxt[user_id].w_id;

        /* 99.6.8 no_range */
    /* check d_id data */
//    if((bp->d_id = str2short (in_data->D_ID, 2)) < 1 || bp->d_id > 10)
    if((bp->d_id = str2short (in_data->D_ID, 2)) < 1 )
        return set_errpage(s_buf, cookie, 2, (int)bp->d_id, 0, 0);

    /* check c_id data */
    if((bp->c_id = str2int (in_data->C_ID, 4)) != -3 ){

//      if (bp->c_id < 1 || bp->c_id > 3000 ){          /*
*/
                if (bp->c_id < 0) {
                        return set_errpage(s_buf, cookie, 6, bp->c_id, 0, 0);
        }
        else{
            c_id_flag = OK;
        }
    }
    else{
        bp->c_id = 0;
    }

    /* check c_last data */
    if((rtn = str2str(in_data->C_LAST, 16)) < 0){
//        return set_errpage(s_buf, cookie, 7, rtn, 22, 0);
/* 99.6.8 no_range              */
                        c_id_flag = OK;    /* 99.6.8 no_range */
    }
    else{
                if ( rtn == 0 || *(in_data->C_LAST) == '\0') {
//                      bp->payin.bylastname = 0;
        /* Oracle use only */
                } else {
                        strcpy (bp->c_last, in_data->C_LAST);
//                      bp->payin.bylastname = 1;
        /* Oracle use only */
                        c_id_flag = OK;
        }
    }

    /* c_id and c_last data is nothing */
    if (c_id_flag == NG)
                return set_errpage(s_buf, cookie, 11, -4, 0, 0);

                /* 99.6.8 no_range */
    /* check c_w_id data */
    if((bp->c_w_id = str2short (in_data->C_W_ID, 4)) < 1 )
//    if((bp->c_w_id = str2short (in_data->C_W_ID, 4)) < 1 || bp->c_w_id > maxwh)
        return set_errpage(s_buf, cookie, 9, (int)bp->c_w_id, 0, maxwh);

                /* 99.6.8 no_range */
    /* check c_d_id data */
    if((bp->c_d_id = str2short (in_data->C_D_ID, 2)) < 1 )
//    if((bp->c_d_id = str2short (in_data->C_D_ID, 2)) < 1 || bp->c_d_id > 10)
        return set_errpage(s_buf, cookie, 10, (int)bp->c_d_id, 0, 0);

    /* check h_amount data :
       str2dbl do hundredfold of the H_ANOUNT. The pupose of if is to prosess
       H_AMOUNT by the integer  : 98.8.3 update */

    if((bp->h_amount = (long)str2dbl (in_data->H_AMOUNT, 7)) < 100 ||
        bp->h_amount > 500000)
        return set_errpage(s_buf, cookie, 8, (int)bp->h_amount, 0, 0);

    DBGP(pay_dsp(in_data, bp, srv->m_tcctxt[user_id].w_id, 0));


/* ---------------------------------------------- Execute Payment transaction */
resend_payment:

#ifndef SCRTEST

    DBGR(tsp(0));

#ifdef USE_FML
        w_id = bp->w_id;

        rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b, FML_TERM, 0, (char *)&w_id, 0 );
        rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b, FML_TRAN, 0, (char *)&bp->tx_type, 0 );
        rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b, FML_DATA, 0, (char *)bp,
                ( FLDLEN )sizeof( payment_trans ) );

    if ( tpcall( TPAPL, ( char * )srv->m_tcctxt[user_id].trans_b, 0,
        ( char ** )&srv->m_tcctxt[user_id].trans_b,
                &olen,0|TPNOTIME ) == -1 ){


        if ( tperrno == TPESVCFAIL ) {
            sprintf( S_WORK, "Oracle failed to process Payment Transaction.\n"
                "tperrno = %d svc = '%s' d_id = %d c_id = %d c_last = '%s'\n"
                                "c_w_id = %d, c_d_id = %d, h_amount = %d\n",
                                tperrno, TPAPL,
                                bp->d_id, bp->c_id, bp->c_last,
                                bp->c_w_id, bp->c_d_id,
                                bp->h_amount );

            set_oraerr( s_buf, S_WORK, cookie );
            return (-1);
        }

                sprintf( S_WORK, "tpcall failed in Payment: tperrno = %d\n"
                        " svc = '%s' d_id = %d c_id = %d c_last = '%s'\n"
                        " c_w_id = %d c_d_id = %d h_amount = %f\n",
                        tperrno, TPAPL,   bp->d_id, bp->c_id, bp->c_last,
                        bp->c_w_id, bp->c_d_id,         bp->h_amount );

                set_tuxerr( s_buf, S_WORK, cookie);
                return (-1);
    }

    DBGR(tsp(1));
        tbuf = *( ( payment_trans * )Ffind( (Fbfr *)srv->m_tcctxt[user_id].trans_b, FML_DATA, 0, NULL ) );
        bp = &tbuf;

#else

    if ( tpcall( TPAPL,(char *)srv->m_tcctxt[user_id].trans_b,
                sizeof(payment_trans),( char ** )&srv->m_tcctxt[user_id].trans_b,
                &olen, 0|TPNOTIME ) == -1 )

        if ( tperrno == TPESVCFAIL ) {
            sprintf( S_WORK, "Oracle failed to process Payment Transaction.\n"
                "tperrno = %d svc = '%s' d_id = %d c_id = %d c_last = '%s'\n"
                                "c_w_id = %d, c_d_id = %d, h_amount = %d\n",
                                tperrno, TPAPL,
                                bp->d_id, bp->c_id, bp->c_last,
                                bp->c_w_id, bp->c_d_id,
                                bp->h_amount );

            set_oraerr( s_buf, S_WORK, cookie );
            return (-1);
        }
```

```
                sprintf( S_WORK, "tpcall failed in Payment: tperrno = %d\n"
                        " svc = '%s' d_id = %d c_id = %d c_last = '%s'\n"
                        " c_w_id = %d c_d_id = %d h_amount = %f\n",
                        tperrno, TPAPL,   bp->d_id, bp->c_id, bp->c_last,
                        bp->c_w_id, bp->c_d_id,         bp->h_amount );

                set_tuxerr( s_buf, S_WORK, cookie);
                return (-1);
    }
    DBGR(tsp(1));

    bp = ( payment_trans * )srv->m_tcctxt[user_id].trans_b;
#endif

/* ---------------------------------------------- Check the execution result */
    if ( bp->C_R != 1){

#ifndef Symfo
                if ( bp->payout.terror == IRRECERR ){
            sprintf (S_WORK, "Irrecoverable error in Payment\n" );
                        set_tuxerr( s_buf, S_WORK, cookie);
                        return (-1);
                }
                goto resend_payment;    /* TP_base busy. Try again */
#else
                set_symfoerr (s_buf, bp->errorpos, bp->sqlstate, cookie) ;
                return (-1);
#endif

    }
#else
    dummy_payment( bp );
#endif

    DBGR(pay_dsp(in_data, bp, srv->m_tcctxt[user_id].w_id, 1));

    sprintf (S_WORK, h_pay2);

#ifdef Symfo
        convert_time (time_data, bp->h_date);
        time2str ((S_WORK + payp[0]), time_data);
#else
        time2str ((S_WORK + payp[0]), bp->h_date );
#endif

    int2str ((S_WORK + payp[1]), 4, (int)bp->w_id);
    int2str ((S_WORK + payp[2]), 2, (int)bp->d_id);
/*
    alp2str ((S_WORK + payp[3]), 20, bp->w_street_1);
    alp2str ((S_WORK + payp[4]), 20, bp->d_street_1);
    alp2str ((S_WORK + payp[5]), 20, bp->w_street_2);
    alp2str ((S_WORK + payp[6]), 20, bp->d_street_2);
*/
        //  check HTML form

        alp2str (&buffer2[0], 20, bp->w_street_1);
        buffer2[20] = 0;
        newlength = checkHTMLform ( &buffer2[0], &buffer[0]);
        strcpy (&buffer3[0], &buffer[0]);
        strcat (buffer3, "                    ");

    alp2str (buffer2, 20, bp->d_street_1);
        buffer2[20] = 0;
        newlength = checkHTMLform ( &buffer2[0], &buffer[0]);
        strcat (buffer3, &buffer[0]);
        strcat (buffer3, "\r\n");

    alp2str (buffer2, 20, bp->w_street_2);
        buffer2[20] = 0;
        newlength = checkHTMLform ( &buffer2[0], &buffer[0]);
        strcat (buffer3, &buffer[0]);
        strcat (buffer3, "                    ");

    alp2str (buffer2, 20, bp->d_street_2);
        buffer2[20] = 0;
        newlength = checkHTMLform ( &buffer2[0], &buffer[0]);
        strcat (buffer3, &buffer[0]);
        strcat (buffer3, "\r\n");

        strcat ( S_WORK, buffer3 );

        // check HTML form

        sprintf ( buffer3, h_pay4 );

    alp2str ((&buffer3[0] + payp[7] - 0xd3), 20, bp->w_city);
    alp2str ((&buffer3[0] + payp[8] - 0xd3), 2,  bp->w_state);
    zip2str ((&buffer3[0] + payp[9] - 0xd3), bp->w_zip);
    alp2str ((&buffer3[0] + payp[11] - 0xd3), 20, bp->d_city);
    alp2str ((&buffer3[0] + payp[12] - 0xd3), 2, bp->d_state);
    zip2str ((&buffer3[0] + payp[13] - 0xd3), bp->d_zip);

    int2str ((&buffer3[0] + payp[15] - 0xd3), 4, bp->c_id);
    int2str ((&buffer3[0] + payp[16] - 0xd3), 4, (int)bp->c_w_id);
    int2str ((&buffer3[0] + payp[17] - 0xd3), 2, (int)bp->c_d_id);

    alp2str ((&buffer3[0] + payp[18] - 0xd3), 16, bp->c_first);
    alp2str ((&buffer3[0] + payp[19] - 0xd3), 2,  bp->c_middle);
    alp2str ((&buffer3[0] + payp[20] - 0xd3), 16, bp->c_last);

#ifdef Symfo
        convert_date (time_data, bp->c_since);
        date2str ((&buffer3[0] + payp[21] - 0xd3), time_data);
#else
        date2str ((S_WORK + payp[21]),bp->c_since);
#endif
/*
    alp2str ((S_WORK + payp[22]), 20, bp->c_street_1);
    alp2str ((S_WORK + payp[23]), 2,  bp->c_credit);

    alp2str ((S_WORK + payp[24]), 20, bp->c_street_2);
        sprintf (buffer3, h_pay5);

  #ifdef Symfo
        dec2str ((S_WORK + payp5[25]), 5,
                (double)((double)(bp->c_discount) / (double)100.0));
#else
        dec2str ((S_WORK + payp[25]), 5, (double)(bp->c_discount * 100.0));
#endif

*/

        strcat (S_WORK, buffer3);

        strcpy (&buffer3[0], "        ");

        alp2str (buffer2, 20, bp->c_street_1);
        buffer2[20] = 0;
        newlength = checkHTMLform ( &buffer2[0], &buffer[0]);
        strcat (buffer3, &buffer[0]);
        strcat (buffer3, "                    Credit: ");

        alp2str (buffer2, 2, bp->c_credit);
        buffer2[2] = 0;
        strcat (buffer3, &buffer2[0]);
        strcat (buffer3, "\r\n");

        strcat (buffer3, "        ");
        alp2str (buffer2, 20, bp->c_street_2);
        buffer2[20] = 0;
        newlength = checkHTMLform ( &buffer2[0], &buffer[0]);
        strcat (buffer3, &buffer[0]);
        strcat (buffer3, "                    %Disc:  ");
        strcat (S_WORK, buffer3);

        dec2str (&buffer3[0], 5,
                (double)((double)(bp->c_discount) / (double)100.0));
        sprintf (&buffer3[5], "\r\n");
        strcat (S_WORK, buffer3);

        sprintf (buffer3, h_pay5);

    alp2str ((&buffer3[0] + payp[26] - 0x21D), 20, bp->c_city);
    alp2str ((&buffer3[0] + payp[27] - 0x21D), 20, bp->c_state);
    zip2str ((&buffer3[0] + payp[28] - 0x21D), bp->c_zip);
    phone2str ((&buffer3[0] + payp[29] - 0x21D), bp->c_phone);

    h_amount = (float)bp->h_amount / (float)100;
    dec2str ((&buffer3[0] + payp[30] - 0x21D), 7, (double)h_amount);

    sigdec2str ((&buffer3[0] + payp[31] - 0x21D), 14, bp->c_balance);
    dec2str ((&buffer3[0] + payp[32] - 0x21D), 13, bp->c_credit_lim);

        strcat (S_WORK, buffer3);

/*
    i = strlen( bp->c_data );
```

```
    alp2str ((S_WORK + payp[33]), 50, bp->c_data);

    if ( i > 50 ){
            alp2str ((S_WORK + payp[34]), 50, &bp-
>c_data[50] );
            if ( i > 100 ){
               alp2str ((S_WORK + payp[35]), 50, &bp-
>c_data[100]);
                    if ( i > 150 ){
                        alp2str ((S_WORK + payp[36]),
50, &bp->c_data[150]);
                    }
            }
    }
*/

    if ( (i = strlen( bp->c_data )) <= 0) {

            sprintf (&buffer3[0], "\r\n\r\n\r\n\r\n\r\n");
        }
    else{
            alp2str (buffer2, 50, bp->c_data);
            buffer2[50] = 0;
            newlength = checkHTMLform ( &buffer2[0],
&buffer[0]);
            strcpy (&buffer3[0], &buffer[0]);
            strcat (buffer3, "\r\n");

            if ( i > 50 ){

                    alp2str (buffer2, 50, &bp-
>c_data[50]);
                    buffer2[50] = 0;
                    newlength = checkHTMLform
( &buffer2[0], &buffer[0]);
                    strcat (buffer3, "          ");
                    strcat (buffer3, &buffer[0]);
                    strcat (buffer3, "\r\n");

                    if ( i > 100 ){

                            alp2str (buffer2, 50, &bp-
>c_data[100]);
                            buffer2[50] = 0;
                            newlength = checkHTMLform
( &buffer2[0], &buffer[0]);
                            strcat (buffer3, "          ");
                            strcat (buffer3, &buffer[0]);
                            strcat (buffer3, "\r\n");

                            if ( i > 150 ){

                                    alp2str (buffer2, 50, &bp-
>c_data[150]);
                                    buffer2[50] = 0;
                                    newlength =
checkHTMLform ( &buffer2[0], &buffer[0]);
                                    strcat (buffer3, "
");
                                    strcat (buffer3,
&buffer[0]);
                                    strcat (buffer3, "\r\n");
                            }
                            else {
                                    strcat ( buffer3,
"\r\n\r\n");
                            }
                    }
                    else {
                            strcat ( buffer3, "\r\n\r\n\r\n");
                    }
            }
        }

        strcat (S_WORK, buffer3);
/* ------- The execuction result data notified RTE is make
by the HTML form */

    sprintf(s_buf, h_pay1);        /* set Header Data */
    strcat (s_buf, S_WORK);    /* set Result Data */

    sprintf(S_WORK, h_pay3, SOPATH, cookie); /* set
Tailer Data */
    strcat (s_buf, S_WORK);

    return (0);
}


/*-------------------------------------------------------------------------
  orderstatus : this function proccesses the Orderstatus
transaction
 -------------------------------------------------------------------------
*/
int orderstatus (char *s_buf, rte_input_data *in_data, int
cookie)
{
    orderstat_trans *bp;
    long olen;
    int  i, user_id, rtn;

        char S_WORK[WORK_S];
        char TPAPL[12];
    char c_id_flag = NG;
        char time_data[64];

#ifdef USE_FML
    orderstat_trans          tbuf;
    int          w_id;
        bp = &tbuf;
        user_id = cookie - Term_Base;
        memset (bp, 0, sizeof(orderstat_trans));
        /* 98.8.29 */
#else
        user_id = cookie - Term_Base;
    bp = ( orderstat_trans * )srv-
>m_tcctxt[user_id].trans_b;
#endif

    bp->tx_type = TX_ORDERSTAT;
        bp->C_R = 0;

#ifdef USE_FML
        sprintf (TPAPL, "TPCC");        // TP-Base
Aplication Name (View)
#else
        sprintf (TPAPL, "TPCC%d", (int)srv-
>m_tcctxt[user_id].trn_id[2] );
#endif

        if ((rtn = ThrTpInit( )) < 0 ){
       sprintf( S_WORK, "Thread init abort ODR (%d) \n",
rtn);
        set_oraerr( s_buf, S_WORK, cookie );
        return (-1);
        }

/* ---------------------------------------------------- check the
Input data */
    bp->w_id = (short)srv->m_tcctxt[user_id].w_id;

                /* 99.6.8 no_range */
    /* check d_id data */
    if ((bp->d_id = str2short (in_data->D_ID, 2)) < 1 )
//    if ((bp->d_id = str2short (in_data->D_ID, 2)) < 1 ||
bp->d_id > 10)
        return set_errpage(s_buf, cookie, 2, (int)bp->d_id,
0, 0);

    /* check c_id data : 98.8.3                    */
    if ((bp->c_id = str2int(in_data->C_ID, 4)) != -3 ){

                DBGP(fprintf (test_fp,    "check c_id=%x\n",
bp->c_id, 0, 0 ));

//              if (bp->c_id < 1 || bp->c_id > 3000 ){ /*
*/
                if (bp->c_id < 0) {
                        return set_errpage(s_buf, cookie, 6,
bp->c_id, 0, 0);
        }
        else{
                        c_id_flag = OK;
                }
    }
    else{
                bp->c_id = 0;
    }

    /* check c_last data */
    if((rtn = str2str(in_data->C_LAST, 16)) < 0){
//        return set_errpage(s_buf, cookie, 7, rtn, 22, 0);
        /* 99.6.8 no_range             */
                c_id_flag = OK;                        /* 99.6.8
no_range      */

    }
    else{
        if ( rtn == 0 || *(in_data->C_LAST) == '\0') {
//                        bp->ordin.bylastname = 0;
         /* Oracle use only */
                } else {
                   strcpy (bp->c_last, in_data->C_LAST);
//                      bp->ordin.bylastname = 1;
         /* Oracle use only */
                        c_id_flag = OK;
        }
    }

    /* c_id and c_last is nothing */
    if (c_id_flag == NG)
        return set_errpage(s_buf, cookie, 11, -4, 0, 0);

 DBGP(oder_dsp ( in_data, bp, srv-
>m_tcctxt[user_id].w_id, 0));

/* ---------------------------------------- Execute Orderstatus
transaction */
resend_orderstatus:

#ifndef SCRTEST

    DBGR(tsp(0));

#ifdef USE_FML
```

```
        w_id = bp->w_id;

        rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b,
FML_TERM, 0, (char *)&w_id, 0 );
        rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b,
FML_TRAN, 0, (char *)&bp->tx_type, 0 );
        rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b,
FML_DATA, 0, (char *)bp,
                ( FLDLEN )sizeof( orderstat_trans ) );

    if ( tpcall( TPAPL, ( char * )srv->m_tcctxt[user_id].trans_b, 0,
        ( char ** )&srv->m_tcctxt[user_id].trans_b,
                &olen, 0|TPNOTIME ) == -1 ){

        if ( tperrno == TPESVCFAIL ) {
            sprintf( S_WORK, "Oracle failed to process
NewOrder Transaction.\n"
                "tperrno = %d svc = '%s' d_id = %d c_id = %d
c_last = '%s'\n",
                tperrno, TPAPL, bp->d_id, bp->c_id,
                bp->c_last );

            set_oraerr( s_buf, S_WORK, cookie );
            return (-1);
        }

                sprintf( S_WORK, "tpcall failed in
OrderStatus: tperrno = %d\n"
                        " svc = '%s' d_id = %d c_id = %d
c_last = '%s'\n",
                        tperrno, TPAPL,bp->d_id, bp->c_id,bp->c_last );

                set_tuxerr (s_buf, S_WORK, cookie);
                return (-1);
    }

    DBGR(tsp(1));
    tbuf = *((orderstat_trans *)Ffind( (Fbfr *)srv->m_tcctxt[user_id].trans_b, FML_DATA, 0, NULL));
        bp = &tbuf;
#else

        if ( tpcall( TPAPL, ( char * )srv->m_tcctxt[user_id].trans_b,
                sizeof(orderstat_trans),
                ( char ** )&srv->m_tcctxt[user_id].trans_b,
&olen, 0|TPNOTIME ) == -1 ){

        if ( tperrno == TPESVCFAIL ) {
            sprintf( S_WORK, "Oracle failed to process
NewOrder Transaction.\n"
                "tperrno = %d svc = '%s' d_id = %d c_id = %d
c_last = '%s'\n",
                tperrno, TPAPL, bp->d_id, bp->c_id,
                bp->c_last );

            set_oraerr( s_buf, S_WORK, cookie );
            return (-1);
        }

                sprintf( S_WORK, "tpcall failed in
OrderStatus: tperrno = %d\n"
                        " svc = '%s' d_id = %d c_id = %d
c_last = '%s'\n",
                        tperrno, TPAPL,bp->d_id, bp->c_id,bp->c_last );

                set_tuxerr (s_buf, S_WORK, cookie);
                return (-1);
    }
    DBGR(tsp(1));

    bp = ( orderstat_trans * )srv->m_tcctxt[user_id].trans_b;

#endif

/* ---------------------------------------- Check the execution
result */
    if ( bp->C_R != 1 ){

#ifndef Symfo
                if ( bp->ordout.terror == IRRECERR ) {

                        sprintf (S_WORK, "Irrecoverable
error in orderstatus.\n" );
                        set_tuxerr (s_buf, S_WORK, cookie);
                        return (-1);
                }

                goto resend_orderstatus;  /* tuxedo busy ->
retry!! */
#else
                set_symfoerr (s_buf, bp->errorpos, bp->sqlstate, cookie) ;
                return (-1);
#endif
    }

#else
    dummy_orderstat( bp );
#endif

    DBGP(oder_dsp ( in_data, bp, srv->m_tcctxt[user_id].w_id, 1));

    sprintf(S_WORK, h_order2);
    int2str ((S_WORK + orderp[0]), 4, (int)bp->w_id);
    int2str ((S_WORK + orderp[1]), 2, (int)bp->d_id);
    int2str ((S_WORK + orderp[2]), 4, bp->c_id);
    alp2str ((S_WORK + orderp[3]), 16,bp->c_first);
    alp2str ((S_WORK + orderp[4]), 2, bp->c_middle);
    alp2str ((S_WORK + orderp[5]), 16,bp->c_last);

    sigdec2str ((S_WORK + orderp[6]), 9, bp->c_balance);

    int2str ((S_WORK + orderp[7]), 8, (int)bp->o_id );

#ifdef Symfo
    convert_time (time_data, bp->o_entry_d);
        time2str ((S_WORK + orderp[8]), time_data);
#else
        time2str ((S_WORK + orderp[8]),bp->o_entry_d );
#endif

    if ( bp->o_carrier_id != INTNULL ) {
                int2str ((S_WORK + orderp[9]), 2, bp->o_carrier_id);
    }

    /* 0x39 is an offset value to the same filed of the next
line */
    for( i = 0; i < bp->o_ol_cnt; i++ ){
        int2str ((S_WORK+i*0x3a+orderp[10]), 4, (int)bp->ol_supply_w_id[i]);
        int2str ((S_WORK+i*0x3a+orderp[11]), 6, (int)bp->ol_i_id[i]);
        int2str ((S_WORK+i*0x3a+orderp[12]), 2, (int)bp->ol_quantity[i]);

#ifdef Symfo
                sigdec2str ((S_WORK+i*0x3a+orderp[13]),
8,
                        (double)((double)bp->ol_amount[i] /
(double)100.0));

//              if( bp->ol_delivery_d[i] != -1 &&
        /* Symfo NG        */
//                      bp->ol_delivery_d[i] != 777777777 ){

                if(     bp->ol_delivery_d[i] != 777777777 ){

                        convert_date (time_data, bp->ol_delivery_d[i]);
                        date2str
((S_WORK+i*0x3a+orderp[14]), time_data);
                }
#else
                sigdec2str ((S_WORK+i*0x3a+orderp[13]),
8,(double)bp->ol_amount[i]);

                if( strncmp( bp->ol_delivery_d[i], "NOT
DELIVR", 10 ) != 0 ){

                        date2str
((S_WORK+i*0x3a+orderp[14]), bp->ol_delivery_d[i]);
                }
#endif
        }

/* ------- The execuction result data notified RTE is make
by the HTML form */

    sprintf(s_buf, h_order1);      /* set Header Data */
    strcat (s_buf, S_WORK);             /* set Result Data
*/

    sprintf (S_WORK, h_order3, SOPATH, cookie); /* set
Tailer Data */
    strcat (s_buf, S_WORK);

    return 0;
}


/*
    delivery : this function proccesses the delivery
transaction.
*/
int delivery (char *s_buf, rte_input_data *in_data, int
cookie)
{
    delivery_trans *bp;
    int         user_id, rtn;
        char    S_WORK[WORK_S];
        char    TPAPL[12];

        struct  tm times;
    SYSTEMTIME systemTime;
```

```
#ifdef USE_FML
        delivery_trans      tbuf;
        int             w_id;

        user_id = cookie - Term_Base;
        bp = &tbuf;

        memset (bp, 0, sizeof(delivery_trans));
        /* 98.7.29 */
#else
//  int  ol_cnt, cnt, i;
//  struct timeval timeque;

        user_id = cookie - Term_Base;
    bp = ( delivery_trans * )srv-
>m_tcctxt[user_id].trans_b;

#endif

        bp->tx_type = TX_DELIVERY;
        bp->C_R = 0;

#ifdef USE_FML
        sprintf (TPAPL, "TPCC");        // TP-Base
Aplication Name (View)
#else
        sprintf (TPAPL, "TPCCd%d", (int)srv-
>m_tcctxt[user_id].trn_id[3] );
#endif

        if ((rtn = ThrTpInit( )) < 0 ){
       sprintf( S_WORK, "Thread init abort DEL(%d) \n",
rtn);
       set_oraerr( s_buf, S_WORK, cookie );
       return (-1);
        }

 /* ------------------------------------------------ Check the Input
data */
    bp->w_id = (short)srv->m_tcctxt[user_id].w_id;

    bp->o_carrier_id = str2short (in_data-
>O_CARRIER_ID, 2);

    if (bp->o_carrier_id < 1 || bp->o_carrier_id > 10)
        /* 98.6.29:             */
       return set_errpage(s_buf, cookie, 5, (int)bp-
>o_carrier_id, 0, 0);

//    bp->delin.in_timing_int = 1;                        /*
oracle use only */

 /* ---------------------------------------- Execute Delivery
transaction */

 resend_delivery:

        GetLocalTime(&systemTime);

  #ifdef  TOOLKIT_ORIGINAL_STRUCTURE              /*
1996.08.07 */
    bp->delin.qtime = ( double )timeque.tv_sec
              + ( double )timeque.tv_usec / 1000000.0;

  #else   /* !TOOLKIT_ORIGINAL_STRUCTURE */

        // GetLocalTIme      ,mktime
        times.tm_year = (int)systemTime.wYear - 1900;
        times.tm_mon  = (int)systemTime.wMonth - 1;
        times.tm_mday = (int)systemTime.wDay;
        times.tm_hour = (int)systemTime.wHour;
        times.tm_min  = (int)systemTime.wMinute;
        times.tm_sec  = (int)systemTime.wSecond;

        bp->startsec  = (long)mktime (&times);
    bp->startusec = (long)systemTime.wMilliseconds;

  #endif  /* !TOOLKIT_ORIGINAL_STRUCTURE */

#ifndef SCRTEST

    DBGR(fprintf ( test_fp, "tpacall delivery!\n" ));

    DBGR(tsp(0));

#ifdef USE_FML
        w_id = bp->w_id;

        rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b,
FML_TERM, 0, (char *)&w_id, 0 );
        rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b,
FML_TRAN, 0, (char *)&bp->tx_type, 0 );
        rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b,
FML_DATA, 0, (char *)bp,
                ( FLDLEN )sizeof( delivery_trans ) );

    rtn = tpacall( TPAPL, ( char * )srv-
>m_tcctxt[user_id].trans_b,
                        0, 0|TPNOREPLY|TPNOTIME ); //
#else
        rtn = tpacall( TPAPL, ( char * )srv-
>m_tcctxt[user_id].trans_b,
                  sizeof( delivery_trans ), 0| TPNOTIME |
TPNOREPLY );
#endif

    DBGR(tsp(1));

#else
    dummy_delivery( bp );
    rtn = 0;
#endif

    sprintf (S_WORK, h_del2);

 /* ------------------------------------ The execution result is
checke. */

    if ( rtn == -1 ) {

    /* Display messege */

#ifndef SCRTEST
        char    buf[1024];

        switch ( tperrno ) {
        case TPELIMIT:  /*                              */
        case TPETIME:   /*                              */
        case TPGOTSIG:  /*                  */

/*          sprintf ( S_WORK, "tpacall : Retry in Delivery:
tperrno = %d\n"
                    " svc = '%s' carrier = %d\n", tperrno,
                    now_ottname, bp->delin.o_carrier_id );
*/
        /* Because it is an executable again error,
           processing is executed again. */
           goto resend_delivery;
           break;

        case TPESVCFAIL:
          sprintf( S_WORK, "Oracle failed to process
Delivery Transaction.\n"
              "tperrno = %d svc = '%s' carrier = %d\n",
tperrno,
                              TPAPL, bp->o_carrier_id );

          set_oraerr( s_buf, S_WORK, cookie );
          return (-1);

        default:
          /* The error which was not able to be executed
again occurred */
           sprintf( S_WORK, "tpacall failed in Delivery:
tperrno = %d\n"
                  " svc = '%s' carrier = %d\n", tperrno,
                  TPAPL, bp->o_carrier_id );

          set_tuxerr (s_buf, S_WORK, cookie);
          return (-1);
        }
#endif

    } else {
                int2str ((S_WORK + delp[0]), 4, (int)bp-
>w_id);
                int2str ((S_WORK + delp[1]), 2, (int)bp-
>o_carrier_id);
                alp2str ((S_WORK + delp[2]), 25, "Delivery
has been queued");
    }

/* -------- The execution result data notified RTE is made
by the HTML form */

    sprintf(s_buf, h_del1);
    strcat (s_buf, S_WORK);

    sprintf(S_WORK, h_del3, SOPATH, cookie);
    strcat (s_buf, S_WORK);

    return 0;
}


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

  stocklevel : this function proccesses the StockLevel
transaction.

  -----------------------------------------------------------------
-*/
int stocklevel (char *s_buf, rte_input_data *in_data, int
cookie)
{
    stocklvl_trans *bp;
    long   olen;
    int loopc = 0;
    int rtn = 0;
        int user_id;

        char S_WORK[WORK_S];
        char TPAPL[12];
```

```
#ifdef USE_FML
    stocklvl_trans   tbuf;
    int          w_id;

    bp = &tbuf;
    user_id = cookie - Term_Base;
        memset (bp, 0, sizeof(stocklvl_trans));
        /* 98.7.29 */
#else
    user_id = cookie - Term_Base;
    bp = ( stocklvl_trans *)srv->m_tcctxt[user_id].trans_b;
#endif

    bp->tx_type = TX_STOCKLVL;
        bp->C_R = 0;

#ifdef USE_FML
        sprintf (TPAPL, "TPCC");        // TP-Base Aplication Name (View)
#else
        sprintf (TPAPL, "TPCCs%d", (int)srv->m_tcctxt[user_id].trn_id[4] );
#endif

        if ((rtn = ThrTpInit( )) < 0 ){
       sprintf( S_WORK, "Thread init abort STOCK (%d)\n", rtn);
        set_oraerr( s_buf, S_WORK, cookie );
        return (-1);
        }

/* ------------------------------------------------------ check the Input data */
    bp->w_id = (short)srv->m_tcctxt[user_id].w_id;
    bp->d_id = (short)srv->m_tcctxt[user_id].d_id;

    /* check threshold data : 98.6.29                  */
    bp->threshold = (long)str2short(in_data->threshold, 2);

    if(bp->threshold < 10 || bp->threshold > 20)
                return set_errpage(s_buf, cookie, 3, (int)bp->threshold, 0, 0);

    DBGP(sto_dsp(in_data, bp, srv->m_tcctxt[user_id].w_id, srv->m_tcctxt[user_id].d_id, 0));


/* ---------------------------------------- Execute Stock Level transaction */
resend_stock:

#ifndef SCRTEST

    DBGR(tsp(0));

 #ifdef USE_FML
        w_id = bp->w_id;

        rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b, FML_TERM, 0, (char *)&w_id, 0);
        rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b, FML_TRAN, 0, (char *)&bp->tx_type, 0);
        rtn = Fchg( (Fbfr *)srv->m_tcctxt[user_id].trans_b, FML_DATA, 0, (char *)bp,
                ( FLDLEN )sizeof( stocklvl_trans ) );

    if ( tpcall( TPAPL, ( char * )srv->m_tcctxt[user_id].trans_b, 0,
         ( char ** )&srv->m_tcctxt[user_id].trans_b,
                &olen, 0|TPNOTIME ) == -1 ){

        if ( tperrno == TPESVCFAIL ) {
            sprintf( S_WORK, "Oracle failed to process StockLevel Transaction.\n"
                "tperrno = %d svc = '%s' threshold = %d\n", tperrno,
                                TPAPL, bp->threshold );

            set_oraerr( s_buf, S_WORK, cookie );
            return (-1);
        }

                sprintf( S_WORK, "stockLevel: tperrno = %d\n"
                        " svc = '%s' threshold = %d\n", tperrno,
                        TPAPL, bp->threshold );

                set_tuxerr (s_buf, S_WORK, cookie);
                return (-1);
    }

    DBGR(tsp(1));
        tbuf = *((stocklvl_trans *)Ffind( (Fbfr *)srv->m_tcctxt[user_id].trans_b, FML_DATA, 0, NULL));
        bp = &tbuf;

 #else

        if ( tpcall( TPAPL, ( char * )srv->m_tcctxt[user_id].trans_b,
                        sizeof( stocklvl_trans ),
            ( char ** )&srv->m_tcctxt[user_id].trans_b, &olen, 0|TPNOTIME ) == -1 ){

        if ( tperrno == TPESVCFAIL ) {
            sprintf( S_WORK, "Oracle failed to process StockLevel Transaction.\n"
                "tperrno = %d svc = '%s' threshold = %d\n", tperrno,
                                TPAPL, bp->threshold );

            set_oraerr( s_buf, S_WORK, cookie );
            return (-1);
        }

                sprintf( S_WORK, "stockLevel: tperrno = %d\n"
                        " svc = '%s' threshold = %d\n", tperrno,
                        TPAPL, bp->threshold );

        set_tuxerr (s_buf, S_WORK, cookie);
        return (-1);
    }
    DBGR(tsp(1));

    bp = ( stocklvl_trans * )srv->m_tcctxt[user_id].trans_b;

 #endif

/* ------------------------------------------ Check the execution result */
    if ( bp->C_R != 1 ){

 #ifndef Symfo
                if ( bp->stoout.terror == IRRECERR ) {
                    sprintf( S_WORK, "Irrecoverable error in stocklevel \n");
                        set_tuxerr (s_buf, S_WORK, cookie);
                    return (-1);
                }
                goto resend_stock;        /* TP application busy. Try again */
 #else
                set_symfoerr (s_buf, bp->errorpos, bp->sqlstate, cookie) ;
                return (-1);
 #endif
    }

#else
    dummy_stocklvl ( bp );
    DBGP(sto_dsp(in_data, bp, srv->m_tcctxt[user_id].w_id, srv->m_tcctxt[user_id].d_id, 1));
#endif

    DBGP(sto_dsp(in_data, bp, srv->m_tcctxt[user_id].w_id, srv->m_tcctxt[user_id].d_id, 1));

    sprintf (S_WORK, h_stock2);
    int2str ((S_WORK + stockp[0]), 4, (int)bp->w_id);
    int2str ((S_WORK + stockp[1]), 2, (int)bp->d_id);
    int2str ((S_WORK + stockp[2]), 2, (int)bp->threshold);
    int2str ((S_WORK + stockp[3]), 3, (int)bp->low_stock);

/* ------- The execuction result data notified RTE is make by the HTML form */

    sprintf(s_buf, h_stock1);     /* Set Header data */
    strcat (s_buf, S_WORK);             /* Set Result data */

    sprintf(S_WORK, h_stock3, SOPATH, cookie); /* Set Tailer data */
    strcat (s_buf, S_WORK);

    return (0);
}



/* ------------------------------------------------------------------
   select_trn:

   RTE


   s_buf                                HTML

   interprets information from the user's input data to determine which
   page should be displayed back to the user.
```

```
   query - the query string that comes back form
ParseFormData
   ptrs  - a pointer to a raw_form_data structure with
pointers
           to values in 'query'.


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

int select_trn ( rte_input_data *in_data, char *s_buf, int
cookie ) {

        int length = 0;
        int user_id = cookie - Term_Base;
        int rtn = 0;

/*
    if(in_data->form) {
*/
    if (in_data->form && (in_data->form[0] != 'M') ) {

                if (in_data->form[0] == 'I' ){
                        /* send the transaction select screen
page */
                        rtn = fast_menu (s_buf, in_data,
cookie);

                        return rtn;
        }
                else{

                        EnterCriticalSection(&(srv-
>m_tcctxt[user_id].user)); // Make it thread safe.

                    /* check transaction type */
                    switch(in_data->form[0]) {
                    case 'N':
                       rtn = neworder (s_buf, in_data,
cookie);
                        break;

             case 'D':
                        rtn = delivery(s_buf, in_data, cookie);
                        break;

                    case 'P':
                           rtn = payment (s_buf, in_data,
cookie);
                        break;

                    case 'S':
                       rtn = stocklevel(s_buf, in_data,
cookie);
                        break;

                         case 'O':
                               rtn = orderstatus (s_buf,
in_data, cookie);
                        break;

                    default:
                               /* uninput transaction type */
                               set_errpage(s_buf, cookie, 1, -
4, 0, 0);
                               rtn = 1;
                        break;
                    }
                        LeaveCriticalSection(&(srv-
>m_tcctxt[user_id].user));
                        return rtn;
                }
    }
    else if(in_data->button) {

                EnterCriticalSection(&(srv-
>m_tcctxt[user_id].user));       // Make it thread safe.

                /* send the data input screen page */
        switch(in_data->button[0]) {
        case 'N':

                        sprintf(s_buf, in_newpage, SOPATH,
cookie, srv->m_tcctxt[user_id].w_id);
                        length = strlen (s_buf);

                        sprintf(s_buf+length -1 ,
in_newpage2);

            break;

        case 'D':
            sprintf(s_buf, in_delpage, SOPATH, cookie, srv-
>m_tcctxt[user_id].w_id);
            break;

        case 'P':
            sprintf(s_buf, in_paypage, SOPATH, cookie,
srv->m_tcctxt[user_id].w_id);
            break;

        case 'S':
            sprintf(s_buf, in_stkpage, SOPATH, cookie,
                                        srv-
>m_tcctxt[user_id].w_id, srv->m_tcctxt[user_id].d_id);
            break;

        case 'O':
            sprintf(s_buf, in_odrpage, SOPATH, cookie,
srv->m_tcctxt[user_id].w_id);
            break;

        case 'Q':
                    /* This value use WWW browser only. */
                        if (in_data->cookie)
                                srv->Termfree (in_data-
>cookie);

            sprintf (s_buf, loginpage , VLDATA, SOPATH);
            return rtn;

        default:
             /* uninput transaction type */
            set_errpage(s_buf, cookie, 0, -4, 0, 0);
            break;
        }

                LeaveCriticalSection(&(srv-
>m_tcctxt[user_id].user));
           return rtn;
    }
    else {

                /* if there is not parameter then send login
page data.
                  this part use WWW browser only */
                sprintf (s_buf, loginpage, VLDATA,
&SOPATH);
                return 0;
    }
}

//////////////////////////////////////////////////////////////
//        CWinApp
//     : MFC    DLL
//

CWinApp theApp;

#ifdef WIZDEF
//////////////////////////////////////////////////////////////
//

BEGIN_PARSE_MAP(CTpaplExtension, CHttpServer)
        // TODO:
ON_PARSE_COMMAND()
        //        ON_PARSE_COMMAND_PARAMS()
        //   :

        ON_PARSE_COMMAND (Default,
CTpaplExtension, ITS_PSTR)
        DEFAULT_PARSE_COMMAND (Default,
CTpaplExtension)
END_PARSE_MAP(CTpaplExtension)
#endif

//////////////////////////////////////////////////////////////
//        CTpaplExtension

CTpaplExtension theExtension;


//////////////////////////////////////////////////////////////
// CTpaplExtension
//
// constructor
//

CTpaplExtension::CTpaplExtension() : CHttpServer('?')
{
        DWORD kind;
        DWORD type;
        DWORD size;

        int   user_id;
        int   def_base = 1;
        int   def_warehouse = 2000;
        int   def_maxusers = 20000;
        int   def_maxterm  = 2000;
        int   CONTROL_Flag = 0;

        int   errcode = 0;

        union dtg {BYTE bit[32]; char chara[32]; DWORD
data[8];} reg_d;

        /* Open registry Area */
        RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"SOFTWARE", 0,
                KEY_READ | KEY_WRITE,
&m_tpccregkey);

        RegCreateKeyEx(m_tpccregkey, "Fujitsu", 0,
NULL,
```

```
            REG_OPTION_NON_VOLATILE,
KEY_READ | KEY_WRITE, NULL,
            &m_tpccregkey, &kind);

      RegCreateKeyEx(m_tpccregkey, "TPC-C ISAPI
Application", 0, NULL,
            REG_OPTION_NON_VOLATILE,
KEY_READ | KEY_WRITE, NULL,
            &m_tpccregkey, &kind);

      /* If the key is newly made,  the default value is
set in the key. */
      if(kind == REG_CREATED_NEW_KEY) {
            RegSetValueEx(m_tpccregkey,
"Term_Base",
                  0, REG_DWORD,(const unsigned
char *)&def_base, 4);
            RegSetValueEx(m_tpccregkey,
"NumWarehouses",
                  0, REG_DWORD,(const unsigned
char *)&def_warehouse, 4);
            RegSetValueEx(m_tpccregkey, "MaxUsers",
                  0, REG_DWORD,(const unsigned
char *)&def_maxusers, 4);
            RegSetValueEx(m_tpccregkey, "MaxTerm
of Client",
                  0, REG_DWORD,(const unsigned
char *)&def_maxterm, 4);
            RegSetValueEx(m_tpccregkey,
"CONTROL_Flag",
                  0, REG_DWORD,(const unsigned
char *)&CONTROL_Flag, 4);
      }

      /* Get registry data */
      RegQueryValueEx(m_tpccregkey, "Term_Base",
            0, &type,(unsigned char *)&reg_d.bit,
&size);
      Term_Base = reg_d.data[0];
      // Start terminal(user) number

      RegQueryValueEx(m_tpccregkey,
"NumWarehouses",
            0, &type,(unsigned char *)&reg_d.bit,
&size);
      maxwh = reg_d.data[0];
      // Max warehouse scale

      RegQueryValueEx(m_tpccregkey, "MaxUsers",
            0, &type,(unsigned char *)&reg_d.bit,
&size);
      maxconnect = reg_d.data[0];
      // Max terminal(user) number

      RegQueryValueEx(m_tpccregkey, "MaxTerm of
Client",
            0, &type,(unsigned char *)&reg_d.bit,
&size);
      maxterm = reg_d.data[0];
      // Max terminal(user) number of client

      RegQueryValueEx(m_tpccregkey,
"CONTROL_Flag",
            0, &type,(unsigned char *)&reg_d.bit,
&size);
      C_FLAG = reg_d.data[0];
            // debug flag

      InitializeCriticalSection (&crit);
      EnterCriticalSection (&crit);


#ifndef SCRTEST

      TLSIsTpInitedKey = TlsAlloc ( );
#endif

      /* Initialize Working Area */
      for(user_id = 0; user_id < maxterm; user_id++) {
            m_tcctxt[user_id].w_id = 0;
            m_tcctxt[user_id].d_id = 0;
            m_tcctxt[user_id].trans_b = 0;
      }

      LeaveCriticalSection (&crit);
      srv = this;

}


//
// deconstructor
//
CTpaplExtension::~CTpaplExtension()
{
      int x;

      EnterCriticalSection(&crit);

      for(x = Term_Base; x < Term_Base + maxterm;
x++){
            Termfree (x);
      }

      LeaveCriticalSection(&crit);

      TlsFree(TLSIsTpInitedKey);
}

BOOL
CTpaplExtension::GetExtensionVersion(HSE_VERSIO
N_INFO* pVer)
{
      //
      CHttpServer::GetExtensionVersion(pVer);

      //
      TCHAR
sz[HSE_MAX_EXT_DLL_NAME_LEN+1];
      ISAPIVERIFY(::LoadString(AfxGetResourceHandl
e(),
                  IDS_SERVER, sz,
HSE_MAX_EXT_DLL_NAME_LEN));
      _tcscpy(pVer->lpszExtensionDesc, sz);
      return TRUE;
}


//////////////////////////////////////////////////////
//  HttpExtensionProc
//////////////////////////////////////////////////////
//  This method is called every time the user makes a
//  request.  We don't do anything here except catch a
successful
//  return value and substitute a value that permits the
//  network connection to be kept alive.

DWORD
CTpaplExtension::HttpExtensionProc(EXTENSION_CO
NTROL_BLOCK *pECB)
{

      char QueryString[1024];
      char *page;
      int  cookie = -1;
      int  len, rtn;

      char S_BUF [BUF_S];
      char S_WORK[WORK_S];

      rte_input_data  in_data_area;

#ifdef DBPRT
   char dbg_query[1024];
   long s_pid;
   char path_w[96];
#endif

      char  szHeader[128];
      char  szHeader1[128];
      int         isize;
      BOOL        rtcode;

      strcpy(QueryString, (char *)pECB-
>lpszQueryString);

#ifdef DBPRT
      strcpy(dbg_query, (char *)pECB-
>lpszQueryString);
#endif

      strct_init ( &in_data_area );

      cookie = anly_para ((char *)QueryString,
&in_data_area );


#ifdef DBPRT
   if ( cookie >= 0){
      sprintf (path_w, "C:\\tclog\\log%d", cookie);
      test_fp = fopen (path_w, "aw+");
   }
#endif

      // Termimal Number Check
      // If terminal number is not valid then send error
message.
      if ( cookie < Term_Base || cookie >= (Term_Base
+ maxterm) ){

            sprintf (S_BUF, badterm, Term_Base,
Term_Base+maxterm-1, cookie);
            len = strlen (S_BUF);

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

```
            isize = sprintf (szHeader, "200 OK");
            sprintf (szHeader1,
                  "Connection: keep-alive\r\nContent-
type: text/html\r\nContent-length: %d\r\n\r\n",
                  len );

            rtcode = (*pECB-
>ServerSupportFunction)(pECB->ConnID,
HSE_REQ_SEND_RESPONSE_HEADER,

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

            rtcode = (*pECB->WriteClient)(pECB-
>ConnID, &S_BUF[0], (unsigned long *)(&len), 0);

            return
HSE_STATUS_SUCCESS_AND_KEEP_CONN;
      }


      //
      rtn = select_trn ( &in_data_area, S_BUF,
cookie );

      //
      len = strlen (S_BUF);

      isize = sprintf (szHeader, "200 OK");

      sprintf (szHeader1,
                  "Connection: keep-alive\r\nContent-
type: text/html\r\nContent-length: %d\r\n\r\n",
                  len );

      rtcode = (*pECB->ServerSupportFunction)(pECB-
>ConnID, HSE_REQ_SEND_RESPONSE_HEADER,

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

      rtcode = (*pECB->WriteClient)(pECB->ConnID,
&S_BUF[0], (unsigned long *)(&len), 0);

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

      // For Debag : output error message
      if ( (C_FLAG & PRT_ELOG) != 0 )
         if ( strstr (S_BUF, "ERROR") != 0) {
               {
               FILE *test_fp2;
               char w_path[96];

            sprintf (w_path, "C:\\tclog\\log%d", cookie);
            test_fp2 = fopen (w_path, "aw+");
                  fprintf (test_fp2, "%s\n", S_BUF);
                  fflush(test_fp2);
                  fclose(test_fp2);
                  }
         }
      }

      return
HSE_STATUS_SUCCESS_AND_KEEP_CONN;

}

////////////////////////////////////////////////////
//  Terminit
////////////////////////////////////////////////////
//  Called when a user logs in.  This function gives the
//  user a cookie and creates a user context in the
server
//  object associated with that cookie.  The context
contains
//  working copies of the response forms with the cookie
//  plugged into them, the warehouse and district ids
//  provided by the user, and the database connection.
//  This function also opens the connection to the
database
//  for this user.

int CTpaplExtension::Terminit(int w_id, int d_id, int
cookie)
{
      int user_id;

# ifdef      USE_FML
#  define     BUF_TYPE "FML"
# else
#  define     BUF_TYPE "CARRAY"
# endif

#ifdef LOG_ALL
      char buffer[80];
#endif LOG_ALL

      // Non-reentrant code.
      // Must find the first free context slot.

      EnterCriticalSection(&crit);    // Make it thread
safe

      user_id = cookie - Term_Base;

      m_tcctxt[user_id].w_id = w_id;
      m_tcctxt[user_id].d_id = d_id;

#ifndef SCRTEST
      // Create Tuxedo buffer (Execute tpalloc)
      if ((m_tcctxt[user_id].trans_b =
            (void *)tpalloc (BUF_TYPE, NULL,
trans_size)) == NULL ) {

            // tpalloc abnormal end. create message
form.

 #ifdef DBPRT
            fprintf (test_fp, "tpalloc( ) faild (%d) :
user_id=%d \n",
                  tperrno, cookie);
 #endif

            LeaveCriticalSection(&crit);
            return (-1);
      }
#else
      // For Debag : create local buffer
      if ((m_tcctxt[user_id].trans_b =
            (void *)malloc ( trans_size )) == NULL ){
            // malloc abnormal end. create message
form.
 #ifdef DBPRT
            fprintf (test_fp, "tpalloc( ) faild : user_id
= %d\n", cookie);
 #endif

            LeaveCriticalSection(&crit);
            return (-1);
      }
#endif

#ifndef USE_FML
      memset ( m_tcctxt[user_id].trans_b, 0,
(size_t)(trans_size));
#endif

      LeaveCriticalSection(&crit);

      InitializeCriticalSection(&(m_tcctxt[user_id].user));

      return cookie;
}



////////////////////////////////////////////////////
//  Termfree
////////////////////////////////////////////////////
//  Given a cookie, determines whether there is a
session
//  associated with it, and if there is, closes the attached
//  database session, frees the associated resources,
and
//  makes the slot available for future use.

void CTpaplExtension::Termfree (int Cookie) {

#ifdef LOG_ALL
      char buffer[80];
#endif LOG_ALL

      int idx = Cookie - Term_Base;

      // Login                        alloc
      if(m_tcctxt[idx].w_id != 0) {

#ifndef SCRTEST
            tpfree ((char *)m_tcctxt[idx].trans_b);
#else
            free (m_tcctxt[idx].trans_b);
#endif

            // Deconstractor           Enter
            LeaveCriticalSection(&(m_tcctxt[idx].user));

            // Login
            m_tcctxt[idx].d_id = 0;
            m_tcctxt[idx].w_id = 0;
      }
}


// For convenience, an overloaded version of Termfree
```

```
// that understands the string representation of the
cookie.
void CTpaplExtension::Termfree (char *Cookie) {

        int x;

        if((x = str2int (Cookie, 4)) < 0) return;
        else Termfree (x);
}


// ClassWizard
#if 0
BEGIN_MESSAGE_MAP(CTpaplExtension,
CHttpServer)
        //{{AFX_MSG_MAP(CTpaplExtension)
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()
#endif         // 0



//////////////////////////////////////////////////////////////////
//                      MFC
//
//                        MFC
// AfxGetResourceHandle()         DllMain()
g_hInstance global
//

/****

static HINSTANCE g_hInstance;

HINSTANCE AFXISAPI AfxGetResourceHandle()
{
        return g_hInstance;
}

BOOL WINAPI DllMain(HINSTANCE hInst, ULONG
ulReason,
                                LPVOID lpReserved)
{
        if (ulReason == DLL_PROCESS_ATTACH)
        {
                g_hInstance = hInst;
        }

        return TRUE;
}

****/
```

File: tpapl.h

```
#if !defined(AFX_TPAPL_H__04D0797B_A452_11D1_8
D77_0000E20BF509__INCLUDED_)
#define
AFX_TPAPL_H__04D0797B_A452_11D1_8D77_0000E
20BF509__INCLUDED_

//    tpapl Extension

#include "resource.h"
#include "tpcciis.h"

int   maxconnect;
int   maxterm;
int   Term_Base;
int   maxwh;
int   C_FLAG;
int   TRN_ID[5];

typedef  struct  tc_tbl {

        void *trans_b;
        int  w_id;
        int  d_id;

        char n_ottname [8];
        char trn_id[5];

        CRITICAL_SECTION user;

} tc_context;

class CTpaplExtension : public CHttpServer
{
public:
        CTpaplExtension();
        ~CTpaplExtension();

        //{{AFX_VIRTUAL(CTpaplExtension)
        public:
        virtual BOOL
GetExtensionVersion(HSE_VERSION_INFO* pVer);
        //}}AFX_VIRTUAL


        DWORD  HttpExtensionProc
( EXTENSION_CONTROL_BLOCK *pECB);
        BOOL ServiceAvailable;

        tc_context m_tcctxt[MAXTERM];

        HKEY m_tpccregkey;


        void Termfree (int Cookie);
        void Termfree (char *Cookie);
        int  Terminit (int w_id, int d_id, int cookie);

        CRITICAL_SECTION crit;

#ifdef WIZDEF
        void Default(CHttpServerContext* pCtxt, LPTSTR
pszName);
        DECLARE_PARSE_MAP()
#endif
        //{{AFX_MSG(CTpaplExtension)
        //}}AFX_MSG
};

CTpaplExtension *srv;

//{{AFX_INSERT_LOCATION}}

#endif
// !defined(AFX_TPAPL_H__04D0797B_A452_11D1_8
D77_0000E20BF509__INCLUDED)
```

File: tpapl.mak

```
# Microsoft Developer Studio Generated NMAKE File,
Based on tpapl.dsp
!IF "$(CFG)" == ""
CFG=tpapl - Win32 keep
!MESSAGE                        tpapl - Win32 keep
!ENDIF

!IF "$(CFG)" != "tpapl - Win32 Release" && "$(CFG)" !=
"tpapl - Win32 keep"
!MESSAGE                        "$(CFG)"
!MESSAGE NMAKE
!MESSAGE
!MESSAGE
!MESSAGE NMAKE /f "tpapl.mak" CFG="tpapl - Win32
keep"
!MESSAGE
!MESSAGE
!MESSAGE
!MESSAGE "tpapl - Win32 Release" ("Win32 (x86)
Dynamic-Link Library"  )
!MESSAGE "tpapl - Win32 keep" ("Win32 (x86)
Dynamic-Link Library"  )
!MESSAGE
!ERROR
!ENDIF

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

!IF  "$(CFG)" == "tpapl - Win32 Release"

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

!IF "$(RECURSE)" == "0"

ALL : "$(OUTDIR)\tpapl.dll"

!ELSE

ALL : "$(OUTDIR)\tpapl.dll"

!ENDIF

CLEAN :
        -@erase "$(INTDIR)\StdAfx.obj"
        -@erase "$(INTDIR)\tpapl.obj"
        -@erase "$(INTDIR)\tpapl.pch"
        -@erase "$(INTDIR)\tpapl.res"
        -@erase "$(INTDIR)\vc50.idb"
        -@erase "$(OUTDIR)\tpapl.dll"
        -@erase "$(OUTDIR)\tpapl.exp"
        -@erase "$(OUTDIR)\tpapl.lib"

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

CPP=cl.exe
CPP_PROJ=/nologo /MD /W3 /GX /O2 /I
"c:\tuxedo\include" /D "WIN32" /D "NDEBUG"\
 /D "_WINDOWS" /D "_WINDLL" /D "_AFXDLL" /D
"_USRDLL" /D "Symfo" /D\
 "_TMSTHREADS" /D "USE_FML"
/Fp"$(INTDIR)\tpapl.pch" /Yu"stdafx.h"\
 /Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /c
```

```
CPP_OBJS=.\Release/
CPP_SBRS=.

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

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

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

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

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

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

MTL=midl.exe
MTL_PROJ=/nologo /D "NDEBUG" /mktyplib203 /o
NUL /win32
RSC=rc.exe
RSC_PROJ=/l 0x411 /fo"$(INTDIR)\tpapl.res" /d
"NDEBUG" /d "_AFXDLL"
BSC32=bscmake.exe
BSC32_FLAGS=/nologo /o"$(OUTDIR)\tpapl.bsc"
BSC32_SBRS= \

LINK32=link.exe
LINK32_FLAGS=libtux.lib libbuft.lib libtux2.lib libfml.lib
libfml32.lib\
 libgp.lib /nologo /subsystem:windows /dll
/incremental:no\
 /pdb:"$(OUTDIR)\tpapl.pdb" /machine:I386
/def:".\tpapl.def"\
 /out:"$(OUTDIR)\tpapl.dll"
/implib:"$(OUTDIR)\tpapl.lib"\
 /libpath:"c:\tuxedo\lib"
DEF_FILE= \
        ".\tpapl.def"
LINK32_OBJS= \
        "$(INTDIR)\StdAfx.obj" \
        "$(INTDIR)\tpapl.obj" \
        "$(INTDIR)\tpapl.res"

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

!ELSEIF  "$(CFG)" == "tpapl - Win32 keep"

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

!IF "$(RECURSE)" == "0"

ALL : "$(OUTDIR)\tpapl.dll" "$(OUTDIR)\tpapl.pch"

!ELSE

ALL : "$(OUTDIR)\tpapl.dll" "$(OUTDIR)\tpapl.pch"

!ENDIF

CLEAN :
        -@erase "$(INTDIR)\StdAfx.obj"
        -@erase "$(INTDIR)\tpapl.obj"
        -@erase "$(INTDIR)\tpapl.pch"
        -@erase "$(INTDIR)\tpapl.res"
        -@erase "$(INTDIR)\vc50.idb"
        -@erase "$(INTDIR)\vc50.pdb"
        -@erase "$(OUTDIR)\tpapl.dll"
        -@erase "$(OUTDIR)\tpapl.exp"
        -@erase "$(OUTDIR)\tpapl.ilk"
        -@erase "$(OUTDIR)\tpapl.lib"
        -@erase "$(OUTDIR)\tpapl.pdb"

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

CPP=cl.exe
CPP_PROJ=/nologo /MDd /W3 /Gm /GX /Zi /Od /I
"c:\tuxedo\include" /D "WIN32" /D\
 "_DEBUG" /D "_WINDOWS" /D "_WINDLL" /D
"_AFXDLL" /D "_USRDLL" /D "Symfo" /D\
 "_TMSTHREADS" /D "USE_FML"
/Fp"$(INTDIR)\tpapl.pch" /Yu"stdafx.h"\
 /Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /c
CPP_OBJS=.\tpaplkeep/
CPP_SBRS=.

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

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

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

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

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

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

MTL=midl.exe
MTL_PROJ=/nologo /D "_DEBUG" /mktyplib203 /o NUL
/win32
RSC=rc.exe
RSC_PROJ=/l 0x411 /fo"$(INTDIR)\tpapl.res" /d
"_DEBUG" /d "_AFXDLL"
BSC32=bscmake.exe
BSC32_FLAGS=/nologo /o"$(OUTDIR)\tpapl.bsc"
BSC32_SBRS= \

LINK32=link.exe
LINK32_FLAGS=libtux.lib libbuft.lib libtux2.lib libfml.lib
libfml32.lib\
 libgp.lib /nologo /subsystem:windows /dll
/incremental:yes\
 /pdb:"$(OUTDIR)\tpapl.pdb" /debug /machine:I386
/def:".\tpapl.def"\
 /out:"$(OUTDIR)\tpapl.dll" /implib:"$(OUTDIR)\tpapl.lib"
/pdbtype:sept\
 /libpath:"c:\tuxedo\lib"
DEF_FILE= \
        ".\tpapl.def"
LINK32_OBJS= \
        "$(INTDIR)\StdAfx.obj" \
        "$(INTDIR)\tpapl.obj" \
        "$(INTDIR)\tpapl.res"

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

!ENDIF


!IF "$(CFG)" == "tpapl - Win32 Release" || "$(CFG)" ==
"tpapl - Win32 keep"
SOURCE=.\StdAfx.cpp
DEP_CPP_STDAF=\
        ".\StdAfx.h"\


!IF  "$(CFG)" == "tpapl - Win32 Release"

CPP_SWITCHES=/nologo /MD /W3 /GX /O2 /I
"c:\tuxedo\include" /D "WIN32" /D\
 "NDEBUG" /D "_WINDOWS" /D "_WINDLL" /D
"_AFXDLL" /D "_USRDLL" /D "Symfo" /D\
 "_TMSTHREADS" /D "USE_FML"
/Fp"$(INTDIR)\tpapl.pch" /Yc"stdafx.h"\
 /Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /c

"$(INTDIR)\StdAfx.obj"  "$(INTDIR)\tpapl.pch" :
$(SOURCE) $(DEP_CPP_STDAF)\
 "$(INTDIR)"
        $(CPP) @<<
  $(CPP_SWITCHES) $(SOURCE)
<<


!ELSEIF  "$(CFG)" == "tpapl - Win32 keep"
```

```
CPP_SWITCHES=/nologo /MDd /W3 /Gm /GX /Zi /Od
/I "c:\tuxedo\include" /D "WIN32"\
 /D "_DEBUG" /D "_WINDOWS" /D "_WINDLL" /D
"_AFXDLL" /D "_USRDLL" /D "Symfo" /D\
 "_TMSTHREADS" /D "USE_FML"
/Fp"$(INTDIR)\tpapl.pch" /Yc"stdafx.h"\
 /Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /c

"$(INTDIR)\StdAfx.obj"  "$(INTDIR)\tpapl.pch" :
$(SOURCE) $(DEP_CPP_STDAF)\
 "$(INTDIR)"
        $(CPP) @<<
  $(CPP_SWITCHES) $(SOURCE)
<<


!ENDIF

SOURCE=.\tpapl.cpp

!IF  "$(CFG)" == "tpapl - Win32 Release"

DEP_CPP_TPAPL=\
        "..\..\tuxedo\include\atmi.h"\
        "..\..\tuxedo\include\fml.h"\
        "..\..\tuxedo\include\tmenv.h"\
        ".\bench2.h"\
        ".\dbgprt.h"\
        ".\delpage.h"\
        ".\dmy.h"\
        ".\fldtbl.h"\
        ".\menupage.h"\
        ".\newpage.h"\
        ".\odrpage.h"\
        ".\paypage.h"\
        ".\stpage.h"\
        ".\tpapl.h"\
        ".\tpcc_info.h"\
        ".\tpcciis.h"\
        ".\tpcinweb.h"\
        ".\tpcweb.h"\
        ".\trans.h"\

CPP_SWITCHES=/nologo /MD /W3 /GX /O2 /I
"c:\tuxedo\include" /D "WIN32" /D\
 "NDEBUG" /D "_WINDOWS" /D "_WINDLL" /D
"_AFXDLL" /D "_USRDLL" /D "Symfo" /D\
 "_TMSTHREADS" /D "USE_FML"
/Fp"$(INTDIR)\tpapl.pch" /Yu"stdafx.h"\
 /Fo"$(INTDIR)\\" /Fd"$(INTDIR)\\" /FD /c

"$(INTDIR)\tpapl.obj" : $(SOURCE)
$(DEP_CPP_TPAPL) "$(INTDIR)"\
 "$(INTDIR)\tpapl.pch"
        $(CPP) @<<
  $(CPP_SWITCHES) $(SOURCE)
<<


!ELSEIF  "$(CFG)" == "tpapl - Win32 keep"

DEP_CPP_TPAPL=\
        "..\..\tuxedo\include\atmi.h"\
        "..\..\tuxedo\include\fml.h"\
        "..\..\tuxedo\include\tmenv.h"\
        ".\bench2.h"\
        ".\dbgprt.h"\
        ".\delpage.h"\
        ".\dmy.h"\
        ".\fldtbl.h"\
        ".\menupage.h"\
        ".\newpage.h"\
        ".\odrpage.h"\
        ".\paypage.h"\
        ".\StdAfx.h"\
        ".\stpage.h"\
        ".\tpapl.h"\
        ".\tpcc_info.h"\
        ".\tpcciis.h"\
        ".\tpcinweb.h"\
        ".\tpcweb.h"\
        ".\trans.h"\

CPP_SWITCHES=/nologo /MDd /W3 /Gm /GX /Zi /Od
/I "c:\tuxedo\include" /D "WIN32"\
 /D "_DEBUG" /D "_WINDOWS" /D "_WINDLL" /D
"_AFXDLL" /D "_USRDLL" /D "Symfo" /D\
 "_TMSTHREADS" /D "USE_FML" /Fo"$(INTDIR)\\"
/Fd"$(INTDIR)\\" /FD /c

"$(INTDIR)\tpapl.obj" : $(SOURCE)
$(DEP_CPP_TPAPL) "$(INTDIR)"
        $(CPP) @<<
  $(CPP_SWITCHES) $(SOURCE)
<<


!ENDIF

SOURCE=.\tpapl.rc

"$(INTDIR)\tpapl.res" : $(SOURCE) "$(INTDIR)"
        $(RSC) $(RSC_PROJ) $(SOURCE)



!ENDIF
```

File: tpcarea.h

```
/* Client context area (oracle web server) */

typedef struct {
//    void *trans_b;    /* pointer of interface area with TP
application */
    int  prt_cnt;             /* print counter : for debug */
    int  clent_num;           /* maximam client matchine
number */
    int  max_user;            /* maximam user number */
    int  ott_num;             /* maximam TP applicaton
program of 1 client */
    int  clent[MAXCLIENT][2];   /* client matchine
infomation */
    int  ott  [MAXCLIENT][MAXOTT]; /* TP application
program information*/
}tpc_struct;
```

File: tpcc_info.h

```
/*=======================================
========================+
  FILENAME : tpcc_info.h
  DESCRIPTION
 +=======================================
========================*/

#ifndef TPCC_INFO_H
#define TPCC_INFO_H

#define trans_size 1104    /* interfase area size */

#ifndef Symfo
/* Oracle use only
  The external variable is declared. (this file use tpcc.c
only) */

long   olen;
void   *trans_buf;
int    trans_size = 1024;
int    svrnum;


int    logincnt = 0;
int    base_cok = 0;
char   NewOrdername[20];
char   Paymentname[20];
char   OrderStatusname[20];
char   Deliveryname[20];
char   StockLevelname[20];


char   s_buf[BUF_S];
char   s_work[WORK_S];

int    now_cookie = 0;
int    now_w_id = 0;
int    now_d_id = 0;
char   now_ottname[8] =
{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00};
#endif


#define VLDATA "Ver 2.1 keep-fml"

#ifdef Symfo
#define INTNULL -32768
#else
#define INTNULL 0
#endif

#ifdef SCRTEST

 #ifdef DBPRT
char   SOPATH[] = "/dbgSD/tpapl.dll";              /*
DEBG Mode: SCRTEST & DBPRT */
#define MDDATA "SCR And DP"
 #else
char   SOPATH[] = "/tpc/tpapl.dll";            /* DEBG
Mode: SCRTEST */
#define MDDATA "SCR"
 #endif

#else

 #ifdef NOSCR
char    SOPATH[] = "/tpc/tpapl.dll";           /* DEGB
Mode: */
#define MDDATA "DBG"
 #else
char    SOPATH[] = "/tpc/tpapl.dll";           /* Release
Mode: */
#define MDDATA "REL"
 #endif
```

```
#endif


#ifdef DBPRT                    /* for debug */
FILE   *test_fp;
#endif

#endif
```

File: tpcciis.h

```
//
// Client Application Header file
//
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <time.h>
#include <math.h>
#include <stdarg.h>
#include <signal.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <malloc.h>
#include <process.h>


/* TPC-C transaction functions
extern int TPCinit ();
extern int TPCnew ();
extern int TPCpay ();
extern int TPCord ();
extern int TPCdel ();
extern int TPCsto ();
extern int TPCexit ();
extern int TPCdumpinit ();
extern int TPCdumpnew ();
extern int TPCdumppay ();
extern int TPCdumpord ();
extern int TPCdumpdel ();
extern int TPCdumpsto ();
extern int TPCdumpexit ();
*/

#define  MAXCONNECT 25000
#define  MAXTERM    6000
#define  MAXWH      2500            /* Max WareHouse
scale */

/* The maximum value of client matchine which can be
processed */
//#define MAXCLIENT  10

/* The maximum value of TP application program of 1
client matchine
   which can be processed */
//#define MAXOTT     80

/* number of Transaction */
#define TRANNEW 1
#define TRANPAY 2
#define TRANORD 3
#define TRANDEL 4
#define TRANSTO 5

/* Error codes : send from TP application program */
#define RECOVERR -10
#define IRRECERR -20
#define NOERR    111

#define NG  0
#define OK  1

#define BUF_S     4096  /* size of the send baffer area
*/
#define WORK_S 2400  /* size of the work buffer area
*/

#define TERM_V   0x1   // Terminal Veriffication
#define PRT_ELOG 0x8// Output Debag Message

// SymfoWare Use Only
#define TX_NEWORDER             1
#define TX_PAYMENT          2
#define TX_ORDERSTAT        3
#define TX_DELIVERY         4
#define TX_STOCKLVL         5


/* Debug Print proc define : debug use only */

#ifdef DBPRT

 #if ( DBPRT > 5 )
#define DBGP(proc)      proc
#define DBGR(proc)      proc

 #else
#define DBGP(proc)
#define DBGR(proc)      proc
 #endif

#else
#define DBGP(proc)
#define DBGR(proc)
#endif
```

File: tpcinweb.h

```
/* ---------------------------------------------------------------------
   tpcinweb.h
   Transaction input data screen data
   ---------------------------------------------------------------------
*/

/* -------------------------
   delivery page
 * -------------------------*/

#define in_delpage "\
<HTML><HEAD><TITLE>TPC-C:
Delivery</TITLE></HEAD>\r\n\
<BODY><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"D\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<center>Delivery<br></center>\r\n\
<font size=4><PRE>Warehouse: %4d\r\n\
\r\n\
Carrier Number: <INPUT NAME=\"OC\" SIZE=2
maxlength=2>\r\n\
\r\n\
Execution Status:\r\n\
</PRE><INPUT
TYPE=\"submit\"></FORM></BODY></HTML>"

#define in_delpage2 "\
<HTML><HEAD><TITLE>TPC-C:
Delivery</TITLE></HEAD>\r\n\
<BODY><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"D\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<center>Delivery<br></center>\r\n\
<font size=3><PRE>\
Warehouse:%4d\r\n\
\r\n\
Carrier Number:<INPUT NAME=\"OC\" SIZE=2
maxlength=2>\r\n\
\r\n\
Execution Status:\r\n\
</PRE><INPUT
TYPE=\"submit\"></FORM></BODY></HTML>\r\n"

/* -------------------------
   neworder page
 * -------------------------*/
#define in_newpage "\
<HTML><HEAD><TITLE>TPC-C: New
Order</TITLE></HEAD>\r\n\
<BODY><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"N\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<center>New Order<br></center>\r\n\
<PRE><font size=4>Warehouse: %4d   District:
<INPUT NAME=\"D\" SIZE=2 maxlength=2>
Date:\r\n\
Customer:  <INPUT NAME=\"CI\" SIZE=4
maxlength=4>  Name:
Credit:     %%Disc:\r\n\
Order Number:          Number of Lines:         W_tax:
D_tax:\r\n\r\n\
 Supp_W  Item_Id  Item Name              Qty  Stock
B/G  Price    Amount\r\n\
  <INPUT NAME=\"OS01\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI01\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ01\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS02\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI02\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ02\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS03\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI03\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ03\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS04\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI04\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ04\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS05\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI05\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ05\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS06\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI06\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ06\" SIZE=2 maxlength=2>\r\n"

#define in_newpage2 "\
  <INPUT NAME=\"OS07\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI07\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ07\" SIZE=2 maxlength=2>\r\n\
```

```
  <INPUT NAME=\"OS08\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI08\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ08\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS09\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI09\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ09\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS10\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI10\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ10\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS11\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI11\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ11\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS12\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI12\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ12\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS13\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI13\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ13\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS14\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI14\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ14\" SIZE=2 maxlength=2>\r\n\
  <INPUT NAME=\"OS15\" SIZE=4 maxlength=4>
<INPUT NAME=\"OI15\" SIZE=6 maxlength=6>
<INPUT NAME=\"OQ15\" SIZE=2 maxlength=2>\r\n\
Execution Status:                                Total:
$\r\n\
</PRE><INPUT
TYPE=\"submit\"></FORM></BODY></HTML>"


/* ---------------------------
   orderstatus page
 * ---------------------------*/
#define in_odrpage "\
<HTML><HEAD><TITLE>TPC-C: Order-
Status</TITLE></HEAD>\r\n\
<BODY><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"O\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<center>Order-Status<br></center>\r\n\
<font size=4><PRE>Warehouse: %4d   District:
<INPUT NAME=\"D\" SIZE=2 maxlength=2>\r\n\
Customer: <INPUT NAME=\"CI\" SIZE=4
maxlength=4>  Name:                <INPUT
NAME=\"CL\" SIZE=17 maxlength=16>\r\n\
Cust-Balance: \r\n\
\r\n\
Order-Number:          Entry-Date:
Carrier-Number:\r\n\
Supply-W    Item-Id   Qty    Amount     Delivery-
Date\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
</PRE>\r\n\
<INPUT
TYPE=\"submit\"></FORM></BODY></HTML>"

#define in_odrpage2 "\
<HTML><HEAD><TITLE>TPC-C: Order-
Status</TITLE></HEAD>\r\n\
<BODY><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"O\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<center>Order Status<br></center>\r\n\
<font size=3><PRE>\
Warehouse:%4d       District:<INPUT NAME=\"D\"
SIZE=2 maxlength=2>\r\n\
Customer:<INPUT NAME=\"CI\" SIZE=4 maxlength=4>
Name:                <INPUT NAME=\"CL\" SIZE=16
maxlength=16>\r\n\
Cust-Balance: \r\n\
\r\n\
Order-Number:          Entry-Date:
Carrier-Number:\r\n\
Supply-W    Item-Id   Qty    Amount     Delivery-
Date\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
</PRE>\r\n\
<INPUT
TYPE=\"submit\"></FORM></BODY></HTML>\r\n"

/* ---------------------------
   payment page
 * ---------------------------*/
#define in_paypage "\
<HTML><HEAD><TITLE>TPC-C:
Payment</TITLE></HEAD>\r\n\
<BODY><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"P\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<center>Payment<br></center>\r\n\
<font size=4><PRE>Date:\r\n\
\r\n\
Warehouse: %4d                District:<INPUT
NAME=\"D\" SIZE=2 maxlength=2>\r\n\
\r\n\
\r\n\
\r\n\
\r\n\
Customer: <INPUT NAME=\"CI\" SIZE=4
maxlength=4>  Cust-Warehouse: <INPUT
NAME=\"CW\" SIZE=4 maxlength=4>  Cust-District:
<INPUT NAME=\"CD\" SIZE=2 maxlength=2>\r\n\
Name:                <INPUT NAME=\"CL\" SIZE=17
maxlength=16>         Since:\r\n\
                                 Credit:\r\n\
                                 %%Disc:\r\n\
                                 Phone:\r\n\
\r\n\
Amount Paid          $<INPUT NAME=\"H\" SIZE=7
maxlength=7>      New Cust-Balance:\r\n\
Credit Limit:\r\n\
\r\n\
Cust-Data:\r\n\
\r\n\
\r\n\
</PRE>\r\n\
<INPUT
TYPE=\"submit\"></FORM></BODY></HTML>"

/* ---------------------------
   stock level page
 * ---------------------------*/
#define in_stkpage "\
<HTML><HEAD><TITLE>TPC-C: Stock-
Level</TITLE></HEAD>\r\n\
<BODY><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"S\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<center>Stock-Level<br></center>\r\n\
<font size=4><PRE>Warehouse: %4d
District: %2d\r\n\
\r\n\
Stock Level Threshold: <INPUT NAME=\"t\" SIZE=2
maxlength=2>\r\n\
\r\n\
low stock:\r\n\
</PRE>\r\n\
<INPUT
TYPE=\"submit\"></FORM></BODY></HTML>"

#define in_stkpage2 "\
<HTML><HEAD><TITLE>TPC-C: Stock-
Level</TITLE></HEAD>\r\n\
<BODY><FORM ACTION=\"%s\"
METHOD=\"GET\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"f\"
VALUE=\"S\">\r\n\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\r\n\
<center>Stock Level<br></center>\r\n\
<font size=3><PRE>\
Warehouse:%4d       District:%2d\r\n\
\r\n\
Stock Level Threshold:<INPUT NAME=\"t\" SIZE=2
maxlength=2>\r\n\
\r\n\
low stock:\r\n\
</PRE>\r\n\
<INPUT
TYPE=\"submit\"></FORM></BODY></HTML>\r\n"
```

File: tpcweb.h

```
/* ---------------------------------------------------------------------
--
   tpcweb.h
   ---------------------------------------------------------------------
--*/

/* If transaction input data is abnormal then use this
format. */
```

```
#define errhtml  "\
<HTML><HEAD><TITLE>ERROR: TPC-
C</TITLE></HEAD><BODY>\
<p>You did something bad.  The error message
was:</p>\
<PRE>%s</PRE>\
<p>Either hit the \"back\" button on your browser and fix
the problem, \
or hit the \"Quit\" button below to terminate this session.
</P><HR>\
<P><FORM ACTION=\"%s\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</FORM></P></BODY></HTML>\r\n"

/* If TP application terminated abnormally then use this
format. */
#define tuxerr "\
<HTML><HEAD><TITLE>ERROR: Tuxedo
</TITLE></HEAD><BODY>\
<P>The database could not process your request.  \
tpcall terminated abnormally.</P>\
<HR><PRE>%s</PRE><HR>\
<FORM ACTION=\"%s\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</BODY></HTML>"

/* If Oracle application terminated abnormally then use
this format. */
#define oraerr "\
<HTML><HEAD><TITLE>ERROR: ORACLE
</TITLE></HEAD><BODY>\
<P>The database could not process your request.  \
Transaction terminated abnormally.</P>\
<HR><PRE>%s</PRE><HR>\
<FORM ACTION=\"%s\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</BODY></HTML>"

/* If SymfoWare application terminated abnormally then
use this format. */
#define symfoerr "\
<HTML><HEAD><TITLE>ERROR:
SYMFOWARE</TITLE></HEAD><BODY>\
<P>The database could not process your request.  \
Transaction terminated abnormally.</P>\
<HR><PRE>%s</PRE><HR>\
<FORM ACTION=\"%s\" METHOD=\"GET\">\
<INPUT TYPE=\"hidden\" NAME=\"c\" VALUE=%d>\
<INPUT TYPE=\"submit\" NAME=\"b\"
VALUE=\"Quit\">\
</BODY></HTML>"

/* If TPINIT( ) abnormally then use this format. */
#define tuxierr "\
<HTML><HEAD><TITLE>ERROR: Tuxedo-init
</TITLE></HEAD><BODY>\
<P>The database could not process your request. \
%s terminated abnormally.</P>\
</BODY></HTML>"
```

File: trans.h

```
/*=======================================
=========================+
 FILENAME : trans.h
 the work struct acoording to transaction is declared.

 +======================================
=========================*/

#ifndef Symfo // if DB-Server is not SymfoWare

/* New order struct */
struct newinstruct {
   int w_id;
   int d_id;
   int c_id;
   int ol_i_id[15];
   int ol_supply_w_id[15];
   int ol_quantity[15];
};

struct newoutstruct {
   int terror;
   int o_id;
   int o_ol_cnt;
   char c_last[17];
   char c_credit[3];
   float c_discount;
   float w_tax;
   float d_tax;
   char o_entry_d[20];
   float total_amount;
   char i_name[15][25];
   int s_quantity[15];
   char brand_generic[15];
   float i_price[15];
   float ol_amount[15];
   char status[26];
   int retry;
};

struct newstruct {
   int tran_kind;
   struct newinstruct newin;
   struct newoutstruct newout;
};


/* Payment struct */
struct payinstruct {
   int w_id;
   int d_id;
   int c_w_id;
   int c_d_id;
   int c_id;
   int bylastname;
/* float h_amount; */ /* old-tool.kit */
   int h_amount;
   char c_last[17];
};

struct payoutstruct {
   int terror;
   char w_street_1[21];
   char w_street_2[21];
   char w_city[21];
   char w_state[3];
   char w_zip[10];
   char d_street_1[21];
   char d_street_2[21];
   char d_city[21];
   char d_state[3];
   char d_zip[10];
   int  c_id;
   char c_first[17];
   char c_middle[3];
   char c_last[17];
   char c_street_1[21];
   char c_street_2[21];
   char c_city[21];
   char c_state[3];
   char c_zip[10];
   char c_phone[17];
   char c_since[11];
   char c_credit[3];
   double c_credit_lim;
   float c_discount;
   double c_balance;
   char c_data[201];
   char h_date[20];
   int retry;
};

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

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

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

struct ordstruct {
   int tran_kind;
   struct ordinstruct ordin;
   struct ordoutstruct ordout;
};


/* Delivery struct */
struct delinstruct {
```

```
    int w_id;
    int o_carrier_id;
    long qtime;
    long uqtime;
    int in_timing_int;
};

struct deloutstruct {
    int terror;
    int retry;
};

struct delstruct {
    int tran_kind;
    struct delinstruct delin;
    struct deloutstruct delout;
};


/* Stock level struct */
struct stoinstruct {
    int w_id;
    int d_id;
    int threshold;
};

struct stooutstruct {
    int terror;
    int low_stock;
    int retry;
};

struct stostruct {
    int tran_kind;
    struct stoinstruct stoin;
    struct stooutstruct stoout;
};

/* Client context area (oracle web server) */

typedef struct {
    void *trans_b;      /* pointer of interface area with TP
application */
    int  prt_cnt;               /* print counter : for debug */
    int  clent_num;             /* maximam client matchine
number */
    int  max_user;              /* maximam user number */
    int  ott_num;               /* maximam TP applicaton
program of 1 client */
    int  clent[MAXCLIENT][2];   /* client matchine
infomation */
    int  ott  [MAXCLIENT][MAXOTT]; /* TP application
program information*/
}tpc_struct;

#endif // IF DB is not SymfoWare.


/* RTE - Client interface struct */
typedef struct {
    char *button,
    *cookie,
    *form,
    *O_CARRIER_ID,
    *threshold,
    *D_ID,
    *C_ID,
    *C_W_ID,
    *C_D_ID,
    *C_LAST,
    *H_AMOUNT,
    *OL_SUPPLY_W_ID[15],
    *OL_I_ID[15],
    *OL_QUANTITY[15];
} rte_input_data;
```

Appendix B: Server Source Code

File: bench1.h

```
/* bench1.h  */

#define  DIST_PER_WARE 10

EXEC SQL BEGIN DECLARE SECTION;
short w_id;
char w_name[11];
char w_street_1[21];
char w_street_2[21];
char w_city[21];
char w_state[3];
char w_zip[10];
long w_tax;
double w_ytd;

short d_id;
char d_name[11];
char d_street_1[21];
char d_street_2[21];
char d_city[21];
char d_state[3];
char d_zip[10];
long d_tax;
long d_ytd; /* add 96.8.13 */
long d_next_o_id;

/*short c_id;*/
/*int c_id; 960823*/
long c_id;
short c_d_id;
short c_w_id;
char c_first[17];
char c_middle[3];
char c_last[17];
char c_street_1[21];
char c_street_2[21];
char c_city[21];
char c_state[3];
char c_zip[10];
char c_phone[17];
/*dtime_t c_since;*/
/*double c_since; 960821*/
/*char c_since[14]; 1997.01.27 */
char c_since[15];
char c_credit[3];
double c_credit_lim;
/*long c_credit_lim;*/
long c_discount;
double c_balance;
/*long c_balance;*/
double c_ytd_payment;
short c_payment_cnt;
/*long c_payment_cnt;*/
char c_data[501];

/*dtime_t h_date;*/
/*double h_date; 960821*/
/*char h_date[14]; 1997.01.27 */
char h_date[15];
long h_amount;
char h_data[25];

long no_o_id;

long o_id;
/*dtime_t o_entry_d;*/
/*double o_entry_d;  960821*/
/*char o_entry_d[14];  1997.01.27 */
char o_entry_d[15];  /*dec     1997.01.27 */
short o_carrier_id;
short o_ol_cnt;
short o_all_local;

long ol_number;
long ol_i_id;
short ol_supply_w_id;
/*dtime_t ol_delivery_d;*/
/*double ol_delivery_d; 960821*/
/*char ol_delivery_d[14]; 1997.01.27 */
char ol_delivery_d[15];
short ol_quantity;
long ol_amount;
/*double ol_amount;*/
char ol_dist_info[25]; /* 1997.01.27 */

long s_quantity;
char s_dist_01[25];    /* 1997.01.27 */
char s_dist_02[25];    /* 1997.01.27 */
char s_dist_03[25];    /* 1997.01.27 */
char s_dist_04[25];    /* 1997.01.27 */
char s_dist_05[25];    /* 1997.01.27 */
char s_dist_06[25];    /* 1997.01.27 */
char s_dist_07[25];    /* 1997.01.27 */
char s_dist_08[25];    /* 1997.01.27 */
char s_dist_09[25];    /* 1997.01.27 */
char s_dist_10[25];    /* 1997.01.27 */
double s_ytd;
long s_order_cnt;
long s_remote_cnt;
char s_data[51];

/*long i_price[15];    */
/*char i_data[15][51]; */
/*char i_name[15][25]; */
long i_priceh;
char i_datah[51];
char i_nameh[25];

EXEC SQL END DECLARE SECTION;
```

File: bench3.h

```
/* ORDERLINE      INSERT   */

   typedef struct{
      long   ol_o_id;
      short  ol_d_id;
      short  ol_w_id;
      long   ol_number;
      long   ol_i_id;
      short  ol_supply_w_id;
/*    char   ol_delivery_d[14];  960912 */
      short  ol_quantity;
/*    char   dummy1[2];          960912 */
      long   ol_amount;
      char   ol_dist_info[25];
      char   dummy2[3];
   }lnk_ol;
```

File: OLCNSERT.scob

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. OLINSERT.
000300 AUTHOR.         H.HARA.
000400 DATE-WRITTEN.   96.08.27.
000500 ENVIRONMENT DIVISION.
000600 CONFIGURATION SECTION.
000900 DATA DIVISION.
001000*
001100 WORKING-STORAGE SECTION.
002200 01  CTR             PIC  S9(04) COMP-5.
002300*
002400     EXEC SQL BEGIN DECLARE SECTION
END-EXEC.
002500 01  G-OL.
002600   02  REC-OL  OCCURS 15.
001500     03  OL-O-ID        PIC  S9(09) COMP-5
SYNC.
001600     03  OL-D-ID        PIC  S9(04) COMP-5
SYNC.
001700     03  OL-W-ID        PIC  S9(04) COMP-5
SYNC.
001800     03  OL-NUMBER      PIC  S9(09) COMP-5
SYNC.
001500     03  OL-I-ID        PIC  S9(09) COMP-5 SYNC.
001600     03  OL-SUPPLY-W-ID PIC  S9(04) COMP-5
SYNC.
001700*     03  OL-DELIVERY-D  PIC  X(14)       .
001800     03  OL-QUANTITY    PIC  S9(04) COMP-5
SYNC.
001700*     03  DUMMY1         PIC  X(02)       .
001700     03  OL-AMOUNT      PIC  S9(09) COMP-5
SYNC.
001800     03  OL-DIST-INFO   PIC  X(24)       .
001700*   03  DUMMY2         PIC  X(04)       .
001000*
004100 01  O-OL-CNT        PIC  S9(04) COMP-5
SYNC.
001000*
004100 01  SQLSTATE        PIC  X(05).
004200 01  SQLMSG          PIC  X(256).
004300     EXEC SQL END   DECLARE SECTION
END-EXEC.
004400*
001100 LINKAGE SECTION.
001200 01  LIN-OL.
001400   02  LIN-REC-OL   OCCURS 15.
001500     03  LIN-OL-O-ID        PIC  S9(09) COMP-5
SYNC.
001600     03  LIN-OL-D-ID        PIC  S9(04) COMP-5
SYNC.
001700     03  LIN-OL-W-ID        PIC  S9(04) COMP-5
SYNC.
001800     03  LIN-OL-NUMBER      PIC  S9(09)
COMP-5 SYNC.
001500     03  LIN-OL-I-ID        PIC  S9(09) COMP-5
SYNC.
001600     03  LIN-OL-SUPPLY-W-ID PIC  S9(04)
COMP-5 SYNC.
001700*     03  LIN-OL-DELIVERY-D  PIC  X(14)       .
```

```
001800     03  LIN-OL-QUANTITY    PIC  S9(04)
COMP-5 SYNC.
001700*     03  LIN-DUMMY1         PIC  X(02)       .
001700     03  LIN-OL-AMOUNT      PIC  S9(09)
COMP-5 SYNC.
001800     03  LIN-OL-DIST-INFO   PIC  X(25)       .
001700     03  LIN-DUMMY2         PIC  X(03)       .
001000*
001400 77  LIN-O-OL-CNT       PIC  S9(04) COMP-5
SYNC.
001000*
004100 77  LIN-SQLSTATE       PIC  X(05).
004500*---------------------------------------------*
004600 PROCEDURE DIVISION USING LIN-OL LIN-
O-OL-CNT LIN-SQLSTATE.
004700*---------------------------------------------*
004800 P-START.
004900     DISPLAY "** OLINSERT START **" UPON
SYSOUT.
005000**   EXEC SQL START SQL END-EXEC.
005100*---------------------------------------------*
006700     INITIALIZE  CTR SQLSTATE.
001500     MOVE LIN-O-OL-CNT TO O-OL-CNT.
005200     PERFORM TEST BEFORE VARYING CTR
FROM 1 BY 1
005200                     UNTIL   CTR > LIN-O-OL-CNT
001500**     MOVE LIN-REC-OL(CTR)  TO REC-
OL(CTR)
001500       MOVE LIN-OL-O-ID(CTR)        TO OL-O-
ID(CTR)
001600       MOVE LIN-OL-D-ID(CTR)        TO OL-D-
ID(CTR)
001700       MOVE LIN-OL-W-ID(CTR)        TO OL-W-
ID(CTR)
001800       MOVE LIN-OL-NUMBER(CTR)      TO OL-
NUMBER(CTR)
001500       MOVE LIN-OL-I-ID(CTR)        TO OL-I-
ID(CTR)
001600       MOVE LIN-OL-SUPPLY-W-ID(CTR) TO
OL-SUPPLY-W-ID(CTR)
001700**     MOVE LIN-OL-DELIVERY-D(CTR)  TO
OL-DELIVERY-D(CTR)
001800       MOVE LIN-OL-QUANTITY(CTR)    TO OL-
QUANTITY(CTR)
001700       MOVE LIN-OL-AMOUNT(CTR)      TO OL-
AMOUNT(CTR)
001800       MOVE LIN-OL-DIST-INFO(CTR)   TO OL-
DIST-INFO(CTR)
011400**     DISPLAY "***** CTR *****  " CTR
018100     DISPLAY "OL-O-ID        = " OL-O-ID(CTR)
018100**     DISPLAY "OL-D-ID        = " OL-D-ID(CTR)
018100**     DISPLAY "OL-W-ID        = " OL-W-
ID(CTR)
018100**     DISPLAY "OL-NUMBER      = " OL-
NUMBER(CTR)
018100**     DISPLAY "OL-I-ID        = " OL-I-ID(CTR)
018100**     DISPLAY "OL-SUPPLY-W-ID = " OL-
SUPPLY-W-ID(CTR)
018100**     DISPLAY "OL-DELIVERY-D  = " OL-
DELIVERY-D(CTR)
018100**     DISPLAY "OL-QUANTITY    = " OL-
QUANTITY(CTR)
018100**     DISPLAY "OL-AMOUNT      = " OL-
AMOUNT(CTR)
018100**     DISPLAY "OL-DIST-INFO   = " OL-DIST-
INFO(CTR)
005200     END-PERFORM.
005100*---------------------------------------------*
015100**   EXEC SQL INSERT INTO
TPCC_SCHEMA.ORDERLINE
015300**                VALUES (:G-OL.REC-OL)
FOR :O-OL-CNT
015100     EXEC SQL INSERT INTO
TPCC_SCHEMA.ORDERLINE(
015100
OL_O_ID,OL_D_ID,OL_W_ID,OL_NUMBER,OL_I_ID,
001600
OL_SUPPLY_W_ID,OL_QUANTITY,OL_AMOUNT,
001800                      OL_DIST_INFO)
015300                VALUES (:G-OL.REC-OL)
FOR :O-OL-CNT
015400     END-EXEC.
015400*
015400     MOVE SQLSTATE TO LIN-SQLSTATE.
015400*
016600     IF SQLSTATE = "00000"
016700       MOVE 0 TO PROGRAM-STATUS
018000     ELSE
016700       MOVE 1 TO PROGRAM-STATUS
011400**     DISPLAY "SQLSTATE  =" SQLSTATE
018100**     DISPLAY "SQLMSG =" SQLMSG(1:256)
017700**     EXEC SQL
017800**             COMMIT  WORK
017900**     END-EXEC
018600     END-IF.
018900*---------------------------------------------*
019000  P-END.
019100**   DISPLAY "** OLINSERT END **".
019200  P-ERR.
019300**   EXEC SQL END SQL END-EXEC.
019400     EXIT PROGRAM.
019400 END PROGRAM OLINSERT.
```

File: stored.h

```
/*---------------------------------------------*/
/*  stored.h : sql declare section for         */
/*             stored proceduer call           */
/*                                             */
/*   1996.9.6 s.sato                           */
/*---------------------------------------------*/
EXEC SQL BEGIN DECLARE SECTION;
char          state[6];
char          sqlmsg[257];
short         sqlmsg_ind;
int           errorpos;

short         w_name_ind;
short         w_street_1_ind;
short         w_street_2_ind;
short         w_city_ind;
short         w_state_ind;
short         w_zip_ind;
short         w_tax_ind;

short         d_id_ind;
short         d_name_ind;
short         d_street_1_ind;
short         d_street_2_ind;
short         d_city_ind;
short         d_state_ind;
short         d_zip_ind;
short         d_tax_ind;

short         c_id_ind;
short         c_first_ind;
short         c_middle_ind;
short         c_last_ind;
short         c_street_1_ind;
short         c_street_2_ind;
short         c_city_ind;
short         c_state_ind;
short         c_zip_ind;
short         c_phone_ind;
short         c_credit_ind;
short         c_credit_lim_ind;
short         c_discount_ind;
short         c_balance_ind;
short         c_ytd_payment_ind;
short         c_payment_cnt_ind;
short         c_since_ind;
varchar       c_datax[501];
short         c_data_ind  ;

short         o_id_ind;
short         o_entry_d_ind;
short         o_carrier_id_ind;
short         o_all_local_ind;

short         no_o_id_ind;

long          ol_i_id1;
long          ol_i_id2;
long          ol_i_id3;
long          ol_i_id4;
long          ol_i_id5;
long          ol_i_id6;
long          ol_i_id7;
long          ol_i_id8;
long          ol_i_id9;
long          ol_i_id10;
long          ol_i_id11;
long          ol_i_id12;
long          ol_i_id13;
long          ol_i_id14;
long          ol_i_id15;
short         ol_i_id1_ind;
short         ol_i_id2_ind;
short         ol_i_id3_ind;
short         ol_i_id4_ind;
short         ol_i_id5_ind;
short         ol_i_id6_ind;
short         ol_i_id7_ind;
short         ol_i_id8_ind;
short         ol_i_id9_ind;
short         ol_i_id10_ind;
short         ol_i_id11_ind;
short         ol_i_id12_ind;
short         ol_i_id13_ind;
short         ol_i_id14_ind;
short         ol_i_id15_ind;
short         ol_supply_w_id1;
short         ol_supply_w_id2;
short         ol_supply_w_id3;
short         ol_supply_w_id4;
short         ol_supply_w_id5;
short         ol_supply_w_id6;
short         ol_supply_w_id7;
short         ol_supply_w_id8;
short         ol_supply_w_id9;
short         ol_supply_w_id10;
short         ol_supply_w_id11;
short         ol_supply_w_id12;
short         ol_supply_w_id13;
```

```
short       ol_supply_w_id14;
short       ol_supply_w_id15;
short       ol_supply_w_id1_ind;
short       ol_supply_w_id2_ind;
short       ol_supply_w_id3_ind;
short       ol_supply_w_id4_ind;
short       ol_supply_w_id5_ind;
short       ol_supply_w_id6_ind;
short       ol_supply_w_id7_ind;
short       ol_supply_w_id8_ind;
short       ol_supply_w_id9_ind;
short       ol_supply_w_id10_ind;
short       ol_supply_w_id11_ind;
short       ol_supply_w_id12_ind;
short       ol_supply_w_id13_ind;
short       ol_supply_w_id14_ind;
short       ol_supply_w_id15_ind;
short       ol_quantity1;
short       ol_quantity2;
short       ol_quantity3;
short       ol_quantity4;
short       ol_quantity5;
short       ol_quantity6;
short       ol_quantity7;
short       ol_quantity8;
short       ol_quantity9;
short       ol_quantity10;
short       ol_quantity11;
short       ol_quantity12;
short       ol_quantity13;
short       ol_quantity14;
short       ol_quantity15;
short       ol_quantity1_ind;
short       ol_quantity2_ind;
short       ol_quantity3_ind;
short       ol_quantity4_ind;
short       ol_quantity5_ind;
short       ol_quantity6_ind;
short       ol_quantity7_ind;
short       ol_quantity8_ind;
short       ol_quantity9_ind;
short       ol_quantity10_ind;
short       ol_quantity11_ind;
short       ol_quantity12_ind;
short       ol_quantity13_ind;
short       ol_quantity14_ind;
short       ol_quantity15_ind;
int         ol_amount1;
int         ol_amount2;
int         ol_amount3;
int         ol_amount4;
int         ol_amount5;
int         ol_amount6;
int         ol_amount7;
int         ol_amount8;
int         ol_amount9;
int         ol_amount10;
int         ol_amount11;
int         ol_amount12;
int         ol_amount13;
int         ol_amount14;
int         ol_amount15;
short       ol_amount1_ind;
short       ol_amount2_ind;
short       ol_amount3_ind;
short       ol_amount4_ind;
short       ol_amount5_ind;
short       ol_amount6_ind;
short       ol_amount7_ind;
short       ol_amount8_ind;
short       ol_amount9_ind;
short       ol_amount10_ind;
short       ol_amount11_ind;
short       ol_amount12_ind;
short       ol_amount13_ind;
short       ol_amount14_ind;
short       ol_amount15_ind;
char        ol_delivery_d1[14];
char        ol_delivery_d2[14];
char        ol_delivery_d3[14];
char        ol_delivery_d4[14];
char        ol_delivery_d5[14];
char        ol_delivery_d6[14];
char        ol_delivery_d7[14];
char        ol_delivery_d8[14];
char        ol_delivery_d9[14];
char        ol_delivery_d10[14];
char        ol_delivery_d11[14];
char        ol_delivery_d12[14];
char        ol_delivery_d13[14];
char        ol_delivery_d14[14];
char        ol_delivery_d15[14];
short       ol_delivery_d1_ind;
short       ol_delivery_d2_ind;
short       ol_delivery_d3_ind;
short       ol_delivery_d4_ind;
short       ol_delivery_d5_ind;
short       ol_delivery_d6_ind;
short       ol_delivery_d7_ind;
short       ol_delivery_d8_ind;
short       ol_delivery_d9_ind;
short       ol_delivery_d10_ind;
short       ol_delivery_d11_ind;
short       ol_delivery_d12_ind;
short       ol_delivery_d13_ind;
short       ol_delivery_d14_ind;
short       ol_delivery_d15_ind;
long        s_quantity1;
long        s_quantity2;
long        s_quantity3;
long        s_quantity4;
long        s_quantity5;
long        s_quantity6;
long        s_quantity7;
long        s_quantity8;
long        s_quantity9;
long        s_quantity10;
long        s_quantity11;
long        s_quantity12;
long        s_quantity13;
long        s_quantity14;
long        s_quantity15;
short       s_quantity1_ind;
short       s_quantity2_ind;
short       s_quantity3_ind;
short       s_quantity4_ind;
short       s_quantity5_ind;
short       s_quantity6_ind;
short       s_quantity7_ind;
short       s_quantity8_ind;
short       s_quantity9_ind;
short       s_quantity10_ind;
short       s_quantity11_ind;
short       s_quantity12_ind;
short       s_quantity13_ind;
short       s_quantity14_ind;
short       s_quantity15_ind;
char        s_dist1[25];
char        s_dist2[25];
char        s_dist3[25];
char        s_dist4[25];
char        s_dist5[25];
char        s_dist6[25];
char        s_dist7[25];
char        s_dist8[25];
char        s_dist9[25];
char        s_dist10[25];
char        s_dist11[25];
char        s_dist12[25];
char        s_dist13[25];
char        s_dist14[25];
char        s_dist15[25];
short       s_dist1_ind;
short       s_dist2_ind;
short       s_dist3_ind;
short       s_dist4_ind;
short       s_dist5_ind;
short       s_dist6_ind;
short       s_dist7_ind;
short       s_dist8_ind;
short       s_dist9_ind;
short       s_dist10_ind;
short       s_dist11_ind;
short       s_dist12_ind;
short       s_dist13_ind;
short       s_dist14_ind;
short       s_dist15_ind;
long        i_priceh1;
long        i_priceh2;
long        i_priceh3;
long        i_priceh4;
long        i_priceh5;
long        i_priceh6;
long        i_priceh7;
long        i_priceh8;
long        i_priceh9;
long        i_priceh10;
long        i_priceh11;
long        i_priceh12;
long        i_priceh13;
long        i_priceh14;
long        i_priceh15;
short       i_priceh1_ind;
short       i_priceh2_ind;
short       i_priceh3_ind;
short       i_priceh4_ind;
short       i_priceh5_ind;
short       i_priceh6_ind;
short       i_priceh7_ind;
short       i_priceh8_ind;
short       i_priceh9_ind;
short       i_priceh10_ind;
short       i_priceh11_ind;
short       i_priceh12_ind;
short       i_priceh13_ind;
short       i_priceh14_ind;
short       i_priceh15_ind;
char        i_nameh1[25];
char        i_nameh2[25];
char        i_nameh3[25];
char        i_nameh4[25];
char        i_nameh5[25];
char        i_nameh6[25];
char        i_nameh7[25];
```

```
char        i_nameh8[25];
char        i_nameh9[25];
char        i_nameh10[25];
char        i_nameh11[25];
char        i_nameh12[25];
char        i_nameh13[25];
char        i_nameh14[25];
char        i_nameh15[25];
short       i_nameh1_ind;
short       i_nameh2_ind;
short       i_nameh3_ind;
short       i_nameh4_ind;
short       i_nameh5_ind;
short       i_nameh6_ind;
short       i_nameh7_ind;
short       i_nameh8_ind;
short       i_nameh9_ind;
short       i_nameh10_ind;
short       i_nameh11_ind;
short       i_nameh12_ind;
short       i_nameh13_ind;
short       i_nameh14_ind;
short       i_nameh15_ind;
char        i_datah1[51];
char        i_datah2[51];
char        i_datah3[51];
char        i_datah4[51];
char        i_datah5[51];
char        i_datah6[51];
char        i_datah7[51];
char        i_datah8[51];
char        i_datah9[51];
char        i_datah10[51];
char        i_datah11[51];
char        i_datah12[51];
char        i_datah13[51];
char        i_datah14[51];
char        i_datah15[51];
short       i_datah1_ind;
short       i_datah2_ind;
short       i_datah3_ind;
short       i_datah4_ind;
short       i_datah5_ind;
short       i_datah6_ind;
short       i_datah7_ind;
short       i_datah8_ind;
short       i_datah9_ind;
short       i_datah10_ind;
short       i_datah11_ind;
short       i_datah12_ind;
short       i_datah13_ind;
short       i_datah14_ind;
short       i_datah15_ind;
int         result_o_id1;
int         result_o_id2;
int         result_o_id3;
int         result_o_id4;
int         result_o_id5;
int         result_o_id6;
int         result_o_id7;
int         result_o_id8;
int         result_o_id9;
int         result_o_id10;
int         result_o_id11;
int         result_o_id12;
int         result_o_id13;
int         result_o_id14;
int         result_o_id15;
short       result_o_id1_ind;
short       result_o_id2_ind;
short       result_o_id3_ind;
short       result_o_id4_ind;
short       result_o_id5_ind;
short       result_o_id6_ind;
short       result_o_id7_ind;
short       result_o_id8_ind;
short       result_o_id9_ind;
short       result_o_id10_ind;
short       result_o_id11_ind;
short       result_o_id12_ind;
short       result_o_id13_ind;
short       result_o_id14_ind;
short       result_o_id15_ind;
short       notfound;
short       notfound_ind;
short       item_notfound;
short       item_notfound_ind;
short       low_stock_ind;
EXEC SQL END DECLARE SECTION;

long    *ol_i_id_str[]      = { (long  *)&ol_i_id1      ,
                              (long  *)&ol_i_id2      ,
                              (long  *)&ol_i_id3      ,
                              (long  *)&ol_i_id4      ,
                              (long  *)&ol_i_id5      ,
                              (long  *)&ol_i_id6      ,
                              (long  *)&ol_i_id7      ,
                              (long  *)&ol_i_id8      ,
                              (long  *)&ol_i_id9      ,
                              (long  *)&ol_i_id10     ,
                              (long  *)&ol_i_id11     ,
                              (long  *)&ol_i_id12     ,
                              (long  *)&ol_i_id13     ,
                              (long  *)&ol_i_id14     ,
                              (long  *)&ol_i_id15     ,
                              NULL};
short   *ol_supply_w_id_str[] = { (short
*)&ol_supply_w_id1 ,
                              (short *)&ol_supply_w_id2 ,
                              (short *)&ol_supply_w_id3 ,
                              (short *)&ol_supply_w_id4 ,
                              (short *)&ol_supply_w_id5 ,
                              (short *)&ol_supply_w_id6 ,
                              (short *)&ol_supply_w_id7 ,
                              (short *)&ol_supply_w_id8 ,
                              (short *)&ol_supply_w_id9 ,
                              (short *)&ol_supply_w_id10,
                              (short *)&ol_supply_w_id11,
                              (short *)&ol_supply_w_id12,
                              (short *)&ol_supply_w_id13,
                              (short *)&ol_supply_w_id14,
                              (short *)&ol_supply_w_id15,
                              NULL};
short   *ol_quantity_str[]  = { (short *)&ol_quantity1  ,
                              (short *)&ol_quantity2  ,
                              (short *)&ol_quantity3  ,
                              (short *)&ol_quantity4  ,
                              (short *)&ol_quantity5  ,
                              (short *)&ol_quantity6  ,
                              (short *)&ol_quantity7  ,
                              (short *)&ol_quantity8  ,
                              (short *)&ol_quantity9  ,
                              (short *)&ol_quantity10 ,
                              (short *)&ol_quantity11 ,
                              (short *)&ol_quantity12 ,
                              (short *)&ol_quantity13 ,
                              (short *)&ol_quantity14 ,
                              (short *)&ol_quantity15 ,
                              NULL};
int     *ol_amount_str[]    = { (int   *)&ol_amount1    ,
                              (int   *)&ol_amount2    ,
                              (int   *)&ol_amount3    ,
                              (int   *)&ol_amount4    ,
                              (int   *)&ol_amount5    ,
                              (int   *)&ol_amount6    ,
                              (int   *)&ol_amount7    ,
                              (int   *)&ol_amount8    ,
                              (int   *)&ol_amount9    ,
                              (int   *)&ol_amount10   ,
                              (int   *)&ol_amount11   ,
                              (int   *)&ol_amount12   ,
                              (int   *)&ol_amount13   ,
                              (int   *)&ol_amount14   ,
                              (int   *)&ol_amount15   ,
                              NULL};
char    *ol_delivery_d_str[]  = { (char
*)&(ol_delivery_d1[0]),
                              (char  *)&ol_delivery_d2   ,
                              (char  *)&ol_delivery_d3   ,
                              (char  *)&ol_delivery_d4   ,
                              (char  *)&ol_delivery_d5   ,
                              (char  *)&ol_delivery_d6   ,
                              (char  *)&ol_delivery_d7   ,
                              (char  *)&ol_delivery_d8   ,
                              (char  *)&ol_delivery_d9   ,
                              (char  *)&ol_delivery_d10  ,
                              (char  *)&ol_delivery_d11  ,
                              (char  *)&ol_delivery_d12  ,
                              (char  *)&ol_delivery_d13  ,
                              (char  *)&ol_delivery_d14  ,
                              (char  *)&ol_delivery_d15  ,
                              NULL};
long    *s_quantity_str[]   = { (long  *)&s_quantity1   ,
                              (long  *)&s_quantity2   ,
                              (long  *)&s_quantity3   ,
                              (long  *)&s_quantity4   ,
                              (long  *)&s_quantity5   ,
                              (long  *)&s_quantity6   ,
                              (long  *)&s_quantity7   ,
                              (long  *)&s_quantity8   ,
                              (long  *)&s_quantity9   ,
                              (long  *)&s_quantity10  ,
                              (long  *)&s_quantity11  ,
                              (long  *)&s_quantity12  ,
                              (long  *)&s_quantity13  ,
                              (long  *)&s_quantity14  ,
                              (long  *)&s_quantity15  ,
                              NULL};
char    *s_dist_str[]       = { (char  *)&(s_dist1[0])  ,
                              (char  *)&s_dist2       ,
                              (char  *)&s_dist3       ,
                              (char  *)&s_dist4       ,
                              (char  *)&s_dist5       ,
                              (char  *)&s_dist6       ,
                              (char  *)&s_dist7       ,
                              (char  *)&s_dist8       ,
                              (char  *)&s_dist9       ,
                              (char  *)&s_dist10      ,
                              (char  *)&s_dist11      ,
                              (char  *)&s_dist12      ,
                              (char  *)&s_dist13      ,
                              (char  *)&s_dist14      ,
                              (char  *)&s_dist15      ,
                              NULL};
```

```
long    *i_priceh_str[]     = { (long  *)&i_priceh1     ,
                        (long  *)&i_priceh2     ,
                        (long  *)&i_priceh3     ,
                        (long  *)&i_priceh4     ,
                        (long  *)&i_priceh5     ,
                        (long  *)&i_priceh6     ,
                        (long  *)&i_priceh7     ,
                        (long  *)&i_priceh8     ,
                        (long  *)&i_priceh9     ,
                        (long  *)&i_priceh10    ,
                        (long  *)&i_priceh11    ,
                        (long  *)&i_priceh12    ,
                        (long  *)&i_priceh13    ,
                        (long  *)&i_priceh14    ,
                        (long  *)&i_priceh15    ,
                        NULL};
char    *i_nameh_str[]      = { (char
*)&(i_nameh1[0])  ,
                        (char  *)&i_nameh2      ,
                        (char  *)&i_nameh3      ,
                        (char  *)&i_nameh4      ,
                        (char  *)&i_nameh5      ,
                        (char  *)&i_nameh6      ,
                        (char  *)&i_nameh7      ,
                        (char  *)&i_nameh8      ,
                        (char  *)&i_nameh9      ,
                        (char  *)&i_nameh10     ,
                        (char  *)&i_nameh11     ,
                        (char  *)&i_nameh12     ,
                        (char  *)&i_nameh13     ,
                        (char  *)&i_nameh14     ,
                        (char  *)&i_nameh15     ,
                        NULL};
char    *i_datah_str[]      = { (char  *)&(i_datah1[0])  ,
                        (char  *)&i_datah2      ,
                        (char  *)&i_datah3      ,
                        (char  *)&i_datah4      ,
                        (char  *)&i_datah5      ,
                        (char  *)&i_datah6      ,
                        (char  *)&i_datah7      ,
                        (char  *)&i_datah8      ,
                        (char  *)&i_datah9      ,
                        (char  *)&i_datah10     ,
                        (char  *)&i_datah11     ,
                        (char  *)&i_datah12     ,
                        (char  *)&i_datah13     ,
                        (char  *)&i_datah14     ,
                        (char  *)&i_datah15     ,
                        NULL};
int     *result_o_id_str[]  = { (int   *)&result_o_id1   ,
                        (int   *)&result_o_id2   ,
                        (int   *)&result_o_id3   ,
                        (int   *)&result_o_id4   ,
                        (int   *)&result_o_id5   ,
                        (int   *)&result_o_id6   ,
                        (int   *)&result_o_id7   ,
                        (int   *)&result_o_id8   ,
                        (int   *)&result_o_id9   ,
                        (int   *)&result_o_id10  ,
                        (int   *)&result_o_id11  ,
                        (int   *)&result_o_id12  ,
                        (int   *)&result_o_id13  ,
                        (int   *)&result_o_id14  ,
                        (int   *)&result_o_id15  ,
                        NULL};

/*-----------------------------------------------*/
/*  stored2.h : sql declare section for       */
/*              stored proceduer call         */
/*                                            */
/*   1996.10.01 s.sato                        */
/*-----------------------------------------------*/
EXEC SQL BEGIN DECLARE SECTION    ;
varchar          s_join[1216]   ;          /* 1997.01.16
*/
short            s_join_ind     ;
varchar          i_join[1216]   ;          /* 1997.01.16
*/
short            i_join_ind     ;
varchar          ol_join[571]   ;
short            ol_join_ind    ;
varchar          ol_q_join[61]  ;
short            ol_q_join_ind  ;
varchar          ol_s_join[61]  ;
short            ol_s_join_ind  ;
varchar          ol_i_join[106] ;
short            ol_i_join_ind  ;
varchar          result_join[101] ;
short            result_join_ind  ;
EXEC SQL END DECLARE SECTION      ;

typedef struct
 {
   short       sqllen          ;
   struct
    {
      /*char     ol_i_id[7]      ;
1997.01.13*/
      char     s_quantity[6]   ;
      char     s_dist[24]      ;
      char     s_data[50]      ;
      char     sapstop[1]      ;
    } sqlvar[15]               ;
 } s_join_str                  ;

typedef struct
 {
   short       sqllen          ;
   struct
    {
      /*char     ol_i_id[7]      ;
1997.01.14*/
      char     i_price[6]      ;
      char     i_name[24]      ;
      char     i_data[50]      ;
      char     sapstop[1]      ;
    } sqlvar[15]               ;
 } i_join_str                  ;

typedef struct
 {
   short       sqllen          ;
   struct
    {
      char     ol_i_id[7]      ;
      char     ol_amount[8]    ;
      char     ol_supply_w_id[4] ;
      char     ol_quantity[4]  ;
      char     ol_delivery_d[14] ;
      char     sapstop[1]      ;
    } sqlvar[15]               ;
 } ol_join_str                 ;

typedef struct                          /* 961003    s.sato
*/
 {
   short       sqllen          ;
   struct
    {
     char     ol_quantity[4]   ;
    } sqlvar[15]               ;
 } ol_q_join_str               ;

typedef struct                          /* 961003    s.sato
*/
 {
   short       sqllen          ;
   struct
    {
     char     ol_supply_w_id[4] ;
    } sqlvar[15]               ;
 } ol_s_join_str               ;

typedef struct                          /* 961003    s.sato
*/
 {
   short       sqllen          ;
   struct
    {
     char     ol_i_id[7]      ;
    } sqlvar[15]               ;
 } ol_i_join_str               ;

typedef struct                          /* 961003    s.sato
*/
 {
   short       sqllen          ;
   struct
    {
     char     result_o_id[9]  ;          /* no_o_id
*/
      char     sapstop[1]      ;
    } sqlvar[10]               ;
 } result_join_str             ;
```

File: TPCC.pc

```
#ifdef NO_SQL
#else
#define USE_SQL_MODE
#endif

/*****************************************************************
********/
/** TPCC.pc     COPYRIGHT FUJITSU LIMITED 1997
**/
/**    :                                              **/
/**    :                                              **/
/**    :       SymfoWARE RDB TPC-C Benchmark
**/
/**    :   Apendix B Server Souurce Code
**/
/**    :       1996/09/06
**/
/**            1997/02/24          (New-order,Order-status)
**/
/**            1997/03/13  Revision3.3 : Any Error(Clause
2.3.6)      **/
/**            1998/03/02  NT      K.Sugiyama & M.Suzuki
**/
/**            1998/06.01  FML       M.Suzuki
**/
```

```
/**                 1998/07/14                  M.Suzuki
**/
/**                 1998/08/21  1000WH      M.Suzuki
**/
/**                 1998/08/27  stocklevel commit
M.Suzuki        **/
/**             1998/11/19                  T.Moriai
**/
/**             1999/05/27      id          K.Serizawa
**/
/*****************************************************************
********/
/*****************************************************************
******************/
/**                                                             **/
/**                     tpccerr     Error()
SQLSTATE            **/
/**                     stderr      Symfo
**/
/*****************************************************************
******************/

#include <windows.h>
#include <sys/types.h>
#include <time.h>
//#include <sys/time.h>
//#include <sys/times.h>
#include <stdio.h>
//#include <sys/param.h>
//#include <sys/ipc.h>
//#include <sys/msg.h>
#include <math.h>
#include <stdlib.h>
//#include <unistd.h>
#include <string.h>

#include "atmi.h"
#include "tmenv.h"
#include "bench2.h"
/*      98.03.02                    nop         */
//#include <userlog.h>
/*      98.03.02        */

#ifdef USE_FML                          /* 98.05.21
*/
# include    "fml.h"
# include    "fldtbl.h"
#endif

#include "bench3.h"                     /*
INSERT 960905 */

/*      98.03.02            */
#define NT
/*              (gettimeofday)      (GetSystemTime)
(SYSTEMTIME)

                                    include  */
#ifdef NT
#include <WTYPES.H>
#endif
/*      98.03.02        */


/*              98.07.01     */
#include <stdio.h>
/* 98.07.01     */

extern void JMPCINT2(),JMPCINT3();              /*
INSERT 960905 */
extern long OLINSETT(lnk_ol *a,short *b,char *c);   /*
INSERT 960905 */

EXEC SQL INCLUDE bench1.h;
EXEC SQL INCLUDE stored.h;                  /*
stored 1996.9.25 sato */
#define INTNULL -32768

/* #define TRACE            on */
#define DP userlog
#define RDB_NORMAL 0

/*      98.03.02            */
/*      98.07.07            */
#ifdef NT
//#define TIMES GetSystemTime(&systemtime);\
//                  tp.tv_sec = ((systemtime.wYear -
1970) *365*24*3600) \
//                              +
((systemtime.wMonth - 1) *30*24*3600    \
//                              +
((systemtime.wDay - 1) *24*3600     \
//                              +
(systemtime.wHour * 3600            \
//                              +
(systemtime.wMinute * 60            \
//                              +
(systemtime.wSecond );\
//                  tp.tv_usec =
systemtime.wMilliseconds * 1000;\
//                  tv_st_sec=tp.tv_sec;
tv_st_usec=tp.tv_usec;
#define TIMES GetSystemTime(&systemtime);\
                    yDay = 0;\
                    switch(systemtime.wMonth -
1){\
                    case 11:    yDay += 30;\
                    case 10:    yDay += 31;\
                    case 9:         yDay += 30;\
                    case 8:         yDay += 31;\
                    case 7:         yDay += 31;\
                    case 6:         yDay += 30;\
                    case 5:         yDay += 31;\
                    case 4:         yDay += 30;\
                    case 3:         yDay += 31;\
                    case 2:         /*      */\
                        (((systemtime.wYear % 4
== 0) && (systemtime.wYear % 100 != 0)) ||\
                        ((systemtime.wYear % 4
== 0) && (systemtime.wYear % 400 == 0)))\
                                ?(yDay +=
29): (yDay += 28);\
                    case 1:         yDay += 31;\
                    default:    break;\
                    }\
                    /*      */\
                    tp.tv_sec =
systemtime.wSecond +
    /*  */\

    (systemtime.wMinute * 60) +
    /*  */\

    ((systemtime.wHour) * 3600) +
        /*  */\

    ((systemtime.wDay - 1) * 3600 * 24);
    /*  */\
                    tp.tv_sec += (yDay * 3600 *
24);                        /*  */\
                    /*              */\
                    yDiff = systemtime.wYear -
1970;       /*    1970 (        ) */\
                    work_day = (yDiff / 4);\
                    tp.tv_sec += ((yDiff * 365) +
work_day) * 24 * 3600;\
                    tp.tv_usec =
systemtime.wMilliseconds;\
                    tv_st_sec=tp.tv_sec;
tv_st_usec=tp.tv_usec;
#else
#define TIMES Gettimeofday(&tp); tv_st_sec=tp.tv_sec;
tv_st_usec=tp.tv_usec;
#endif

#ifdef NT
#define TIMEE(NUM) \
//      GetSystemTime(&systemtime);\
//          tp.tv_sec = ((systemtime.wYear - 1970)
*365*24*3600) \
//                      + ((systemtime.wMonth - 1)
*30*24*3600    \
//                      + ((systemtime.wDay - 1)
*24*3600       \
//                      + (systemtime.wHour * 3600
\
//                      + (systemtime.wMinute * 60
\
//                      + (systemtime.wSecond );\
            GetSystemTime(&systemtime);\
            yDay = 0;\
            switch(systemtime.wMonth - 1){\
            case 11:    yDay += 30;\
            case 10:    yDay += 31;\
            case 9:         yDay += 30;\
            case 8:         yDay += 31;\
            case 7:         yDay += 31;\
            case 6:         yDay += 30;\
            case 5:         yDay += 31;\
            case 4:         yDay += 30;\
            case 3:         yDay += 31;\
            case 2:         /*      */\
                    (((systemtime.wYear % 4 == 0) &&
(systemtime.wYear % 100 != 0)) ||\
                     ((systemtime.wYear % 4 == 0) &&
(systemtime.wYear % 400 == 0)))\
                            ?(yDay += 29): (yDay +=
28);\
            case 1:         yDay += 31;\
            default:    break;\
            }\
            /*      */\
            tp.tv_sec = systemtime.wSecond +
                            /*  */\
                            (systemtime.wMinute *
60) +                       /*  */\
                            ((systemtime.wHour) *
3600) +                         /*  */\
                            ((systemtime.wDay - 1) *
3600 * 24);             /*  */\
            tp.tv_sec += (yDay * 3600 * 24);
                        /*  */\
            /*                  */\
```

```
            yDiff = systemtime.wYear - 1970;
        /*      1970  (         ) */\
            work_day = (yDiff / 4);\
            tp.tv_sec += ((yDiff * 365) + work_day) * 24
* 3600;\
            tp.tv_usec = systemtime.wMilliseconds;\
        time_sec=tp.tv_sec-tv_st_sec;\
        if(tp.tv_usec < tv_st_usec) \
        {       time_usec=1000-tv_st_usec+tp.tv_usec;
time_sec=time_sec-1; } \
        else \
        time_usec=tp.tv_usec-tv_st_usec; \
        time_usec=time_sec*1000+time_usec; \
        if(NUM!=999) \
        {       fprintf(time_fd,"SQL_NUM = %d
EACH_TIME= %d\n",NUM,time_usec); \
            all_time(NUM,time_sec,time_usec); \
        } \
        else \
        {       fprintf(time_fd,"ALL_NUM = %d
EACH_TIME= %d.%06d\n", \
                                            NUM,
time_sec, time_usec ); \
        }
#else
#define TIMEE(NUM) \
        Gettimeofday(&tp);\
        time_sec=tp.tv_sec-tv_st_sec;\
        if(tp.tv_usec < tv_st_usec) \
        {       time_usec=1000000-tv_st_usec+tp.tv_usec;
time_sec=time_sec-1; } \
        else \
        time_usec=tp.tv_usec-tv_st_usec; \
        time_usec=time_sec*1000000+time_usec; \
        if(NUM!=999) \
        {       fprintf(time_fd,"SQL_NUM = %d
EACH_TIME= %d\n",NUM,time_usec); \
            all_time(NUM,time_sec,time_usec); \
        } \
        else \
        {       fprintf(time_fd,"ALL_NUM = %d
EACH_TIME= %d.%06d\n", \
                                            NUM,
time_sec, time_usec ); \
        }
#endif
/*      98.03.02        */

/*#define SOLARIS       */      /* 98.02.23  suzuki */

#ifdef UXP_DS  /* 98.02.23     suzuki */
#define Gettimeofday(a) gettimeofday(a)
/*      98.03.02                */
#elif defined NT
//#define Gettimeofday(a)
GetSystemTime(&systemtime);\
//                                      *a.tv_sec =
((systemtime.wYear - 1970) *365*24*3600) \
//                                              +
((systemtime.wMonth - 1) *30*24*3600)    \
//                                              +
((systemtime.wDay - 1) *24*3600)         \
//                                              +
(systemtime.wHour * 3600)                \
//                                              +
(systemtime.wMinute * 60)                \
//                                              +
(systemtime.wSecond );\
//                                      *a.tv_usec =
systemtime.wMilliseconds * 1000;
#define Gettimeofday(a)
GetSystemTime(&systemtime);\
                                        yDay = 0;\
        switch(systemtime.wMonth - 1){\
                                        case 11:     yDay
+= 30;\
                                        case 10:     yDay
+= 31;\
                                        case 9:
        yDay += 30;\
                                        case 8:
        yDay += 31;\
                                        case 7:
        yDay += 31;\
                                        case 6:
        yDay += 30;\
                                        case 5:
        yDay += 31;\
                                        case 4:
        yDay += 30;\
                                        case 3:
        yDay += 31;\
                                        case 2:\
                                            /*          */\

        (((systemtime.wYear % 4 == 0) &&
(systemtime.wYear % 100 != 0)) ||\

((systemtime.wYear % 4 == 0) &&
(systemtime.wYear % 400 == 0)))\

        ?(yDay += 29): (yDay += 28);\
                                        case 1:
        yDay += 31;\
                                        default:       break;\
                                        }\
                                        *a.tv_sec =
systemtime.wSecond +
        /*    */\

        (systemtime.wMinute * 60) +
        /*    */\

        ((systemtime.wHour) * 3600) +
                /*    */\

        ((systemtime.wDay - 1) * 3600 * 24);
        /*    */\
                                        *a.tv_sec += (yDay
* 3600 * 24);                                  /*    */\
                                        /*          */\
                                        /*
*/\
                                        yDiff =
systemtime.wYear - 1970;                /*       1970
(         ) */\
                                        work_day = (yDiff /
4);\
                                        *a.tv_sec += ((yDiff
* 365) + work_day) * 24 * 3600;\
                                        *a.tv_usec =
systemtime.wMilliseconds;
/*      98.03.02        */
#else
#define Gettimeofday(a) gettimeofday(a,0)
#endif



/* Function Prototype */
extern int  scanstring();

/* add-96.8.23 */
time_t  tttt;
time_t  t_wk;
char    tc_wk[26];
char    tc_s[15]; /* 1997.01.27 */

#ifdef USE_SQL_MODE                     /* 98.02.23
suzuki */
EXEC SQL BEGIN DECLARE SECTION;
short errorpos_ind;
#endif
int tmp_s_i_id;
int tmp_w_id;
int tmp_d_id;

long namecount;

long ol_total;

long low_stock;
long threshold;
int  tmp_o_id;
char SQLSTATE[6];

/* 98.06.08                     */
int
t19,t18,t17,t16,t15,t14,t13,t12,t11,t10,t09,t08,t07,t06,t0
5,t04,t03,t02;
/* 98.06.08     */

#ifdef USE_SQL_MODE                     /* 98.02.23
suzuki */

EXEC SQL END DECLARE SECTION;

#else

#define OLINSERT OLINSERT_nop
#define JMPCINT2 JMPCINT2_nop
#define JMPCINT3 JMPCINT3_nop
OLINSERT_nop(){}
JMPCINT2_nop(){}
JMPCINT3_nop(){}

#ifdef NT
#define       SQLWAIT_O   Sleep( 1 );
#define       SQLWAIT_N   Sleep( 1 );
#define       SQLWAIT_N_C Sleep( 1 );
#define       SQLWAIT_N_R Sleep( 2 );
#define       SQLWAIT_P   Sleep( 1 );
#define       SQLWAIT_D   Sleep( 5 );
#define       SQLWAIT_S   Sleep( 2 );
#else
#define SLEEP_MIN 10
#define       SQLWAIT_O   usleep(     10 *
SLEEP_MIN);
#define       SQLWAIT_N   usleep(    100 *
SLEEP_MIN);
#define       SQLWAIT_N_C usleep(     10 *
SLEEP_MIN);
```

```
#define       SQLWAIT_N_R usleep(    200 *
SLEEP_MIN);
#define       SQLWAIT_P   usleep(     20 *
SLEEP_MIN);
#define       SQLWAIT_D   usleep(    500 *
SLEEP_MIN);
#define       SQLWAIT_S   usleep(    200 *
SLEEP_MIN);
#endif

#endif

neworder_trans  *bpn;
payment_trans   *bpp;
orderstat_trans *bpo;
delivery_trans  *bpd;
stocklvl_trans  *bps;

#ifdef USE_FML                                  /* 98.05.21
*/
neworder_trans  nbuf;
payment_trans   pbuf;
orderstat_trans obuf;
delivery_trans  dbuf;
stocklvl_trans  sbuf;
#endif

/*      98.03.02              */
/*                      (gettimeofday)                  */
#ifdef NT
struct _SYSTEMTIME systemtime;
struct tp_tag{
        long  tv_sec       ;
        long  tv_usec      ;
      } ;
struct tp_tag tp,tp_e;
#else
struct timeval tp,tp_e;
#endif
/*      98.03.02        */
long tv_st_sec,tv_st_usec;
long time_sec,time_usec;
/* long tv_en_sec,tv_en_usec;  for delivery */
long result_o_id[10];      /* for delivery */
int number;
int douitu;

FILE *fd = 0;
FILE *t_fd = 0;
FILE *time_fd ;

FILE    *delivery_handle = NULL;
FILE    *fp;


/*              98.07.01      */
        static FILE *tpsvrinit_fp = 0;
        static ctr = 0;
/* 98.07.01       */

void    s_ymdhms()
{
       struct tm       tim;

       time(&t_wk)                  ;
       tim = *( localtime( &t_wk ) ) ;
       sprintf( tc_s, "%04d%02d%02d%02d%02d%02d",
                tim.tm_year + 1900, tim.tm_mon+1,
tim.tm_mday,
                tim.tm_hour, tim.tm_min, tim.tm_sec ) ;
/*       tc_s[14] = NULL ; */
       tc_s[14] = 0 ;
}

long    c_ymdhms( char *time )
{
       struct  tm  itm              ;
       long        otm              ;
       int         ymdhms           ;
       char        ctm[3]           ;

       ctm[2]      = '\0'           ;

       strncpy( ctm , &time[2] , 2 ) ;
       ymdhms      =  atoi( ctm )   ;
       itm.tm_year  =  ymdhms       ;

       strncpy( ctm , &time[4] , 2 ) ;
       ymdhms      =  atoi( ctm )   ;
       itm.tm_mon   =  ymdhms - 1    ;

       strncpy( ctm , &time[6] , 2 ) ;
       ymdhms      =  atoi( ctm )   ;
       itm.tm_mday  =  ymdhms       ;

       strncpy( ctm , &time[8] , 2 ) ;
       ymdhms      =  atoi( ctm )   ;
       itm.tm_hour  =  ymdhms       ;

       strncpy( ctm , &time[10], 2 ) ;
       ymdhms      =  atoi( ctm )   ;
       itm.tm_min   =  ymdhms       ;

       strncpy( ctm , &time[12], 2 ) ;
       ymdhms      =  atoi( ctm )   ;
       itm.tm_sec   =  ymdhms       ;

       itm.tm_isdst = -1            ;

       otm = mktime( &itm )         ;
       return( otm )                ;
}

/************************************************/
/*  TPCC                                        */
/************************************************/
int TPCC(info,num)
TPSVCINFO  *info;
int  num;
{
       int     mix;
       int     k;
       char    logname[80]; /* for delivery  1997.02.27 */

       FILE *fp;
       int  rtnsize;

/* 98.07.07          */
                DWORD
        work,yDay,yDiff,work_day;
/* 98.07.07      */

       /* wait for message to come in */
#ifdef USE_FML                                  /* 98.04.09
Ich. */
       mix = Fvall( ( FBFR * )info->data, FML_TRAN,
0 );
#else
       mix = *((int *)info->data);
#endif
#ifdef  TRACE
       DP("TPCC-call mix=%d \n",mix);
#endif

       if( mix == 1 )
       {
#ifdef USE_FML                                  /* 98.04.07
Ich. */
          nbuf = *( ( neworder_trans * )Ffind( ( FBFR
* )info->data,
              FML_DATA, 0, NULL ) );
          bpn = &nbuf;
#else
          bpn = (neworder_trans *)info->data;
#endif
          rtnsize = sizeof(neworder_trans);
          w_id = bpn->w_id ;
          d_id = bpn->d_id ;
          bpn->C_R = 0;          /* Commit/Rollback flag */
          tmp_d_id = bpn->d_id;
          c_id = bpn->c_id;
          o_ol_cnt = bpn->o_ol_cnt;

          bpn->C_R = NewOrder();
#ifdef USE_FML                                  /* 98.04.07
Ich. */
          Fchg( ( FBFR * )info->data, FML_DATA, 0,
( char * )bpn,
              sizeof( neworder_trans ) );
#endif
       }
       else if( mix == 2 )
       {
#ifdef USE_FML                                  /* 98.04.07
Ich. */
          pbuf = *( ( payment_trans * )Ffind( ( FBFR
* )info->data,
              FML_DATA, 0, NULL ) );
          bpp = &pbuf;
#else
          bpp = (payment_trans *)info->data;
#endif
          rtnsize = sizeof(payment_trans);
          w_id = bpp->w_id ;
          d_id = bpp->d_id ;

          c_d_id = bpp->c_d_id;
          c_w_id = bpp->c_w_id;

          strncpy(c_last," ",sizeof(c_last));          /*
960909 */
          strcpy(c_last,bpp->c_last);
          for(k=0 ; k<16; k++)
          {
             if (c_last[k] == 0x00)
                          {
                                c_last[k] = 0x20;
                          }
                   }
          c_id = bpp->c_id;
          h_amount = bpp->h_amount;
```

```
        bpp->C_R = Payment();
#ifdef USE_FML                          /* 98.04.07
Ich. */
         Fchg( ( FBFR * )info->data, FML_DATA, 0,
( char * )bpp,
            sizeof( payment_trans ) );
#endif
      }
      else if( mix == 3 )
      {
#ifdef USE_FML                          /* 98.04.07
Ich. */
         obuf = *( ( orderstat_trans * )Ffind( ( FBFR
* )info->data,
            FML_DATA, 0, NULL ) );
         bpo = &obuf;
#else
        bpo = (orderstat_trans*)info->data;
#endif
        rtnsize = sizeof(orderstat_trans);
        w_id = bpo->w_id ;
        d_id = bpo->d_id ;
        c_id = bpo->c_id;
        bpo->C_R = 0;   /* Clear the Commit/Rollback
flag */
        c_w_id = bpo->w_id;     /* clients Warehouse ID
*/
        c_d_id =bpo->d_id;

        strncpy(c_last," ",sizeof(c_last));             /*
960909 */
        strcpy(c_last,bpo->c_last);
        for(k=0 ; k<16; k++)
        {
           if (c_last[k] == 0x00)
                        {
                                c_last[k] = 0x20;
                        }
                }

        if(OrderStatus())
        {           bpo->C_R = 1; }
        else
        {           bpo->C_R = 0; }
#ifdef USE_FML                          /* 98.04.07
Ich. */
         Fchg( ( FBFR * )info->data, FML_DATA, 0,
( char * )bpo,
            sizeof( orderstat_trans ) );
#endif
      }
      else if( mix == 4 )
      {
#ifdef USE_FML                          /* 98.04.07
Ich. */
         dbuf = *( ( delivery_trans * )Ffind( ( FBFR
* )info->data,
            FML_DATA, 0, NULL ) );
         bpd = &dbuf;
#else
        bpd = (delivery_trans*)info->data;
#endif

        w_id = bpd->w_id ;
        d_id = bpd->d_id ;
        bpd->C_R = 0;           /* Clear the
Commit/Rollback flag */
        if ( delivery_handle == NULL )
        {
#ifdef USE_FML                          /* 98.04.07
Ich. */
                num = getpid();         /* (^^; */
#endif
                sprintf(logname,"/tpcrt/tpcc/delivery_log%d
",num);
                delivery_handle = fopen(logname,"w+");
                if ( delivery_handle == NULL )
                {       delivery_handle = stderr;
#ifdef  TRACE
                        DP("delivery_log1 cannot write\n");
#endif
                        printf("delivery_log cannot write\n");
                        fflush(stdout);
                }
        }
        o_carrier_id = bpd->o_carrier_id;

        for(d_id = 0; d_id < 10; d_id++)
        {       result_o_id[d_id ] = 0;         }

        /********** Transaction ***********/
        bpd->C_R = Delivery();
        Gettimeofday(&tp_e);
#ifdef  TRACE
        DP("A-deli_handle= %x C_R=%d
\n",delivery_handle,bpd->C_R);
#endif
        if(bpd->C_R)
        {

#ifdef NT
/* NT      msec                    1000            */
          fprintf(delivery_handle,"%09d%03d %09d%03
d %d %d",
                bpd->startsec,
                bpd->startusec,
                tp_e.tv_sec,
                tp_e.tv_usec,
                w_id,
                o_carrier_id);
#else
          fprintf(delivery_handle,"%09d%03d %09d%03
d %d %d",
                bpd->startsec,
                bpd->startusec/1000,
                tp_e.tv_sec,
                tp_e.tv_usec/1000,
                w_id,
                o_carrier_id);
#endif

          for(d_id = 0; d_id < 10; d_id++)
          {
             fprintf(delivery_handle," %d %d",d_id+1,re
sult_o_id[d_id]);
          }
          fprintf(delivery_handle, "\n");
        }
        else
        {

#ifdef NT
/* NT      msec                    1000            */
          fprintf(delivery_handle,"%09d%03d %09d%03
d %d %d",
                bpd->startsec,
                bpd->startusec,
                0,
                0,
                w_id,
                o_carrier_id);
#else
          fprintf(delivery_handle,"%09d%03d %09d%03
d %d %d",
                bpd->startsec,
                bpd->startusec/1000,
                0,
                0,
                w_id,
                o_carrier_id);
#endif
          fprintf(delivery_handle," errpos:%04d
SQLSTATE:%05d\n",
                        bpd->errorpos,bpd-
>sqlstate);
        }
#ifdef  TRACE
      DP("Out-deli_handle= %x C_R=%d
\n",delivery_handle,bpd->C_R);
#endif
#ifdef  TRACE
   DP("tpreturn-called MIX =%d TPNOREPLY \n",mix);
#endif
  /*tpreturn(TPSUCCESS,0,(char
*)bpd,sizeof(delivery_trans),0|TPNOREPLY);*/
#ifdef USE_FML                          /* 98.04.07
Ich. */
          tpreturn( TPSUCCESS, 0, (char *)NULL, 0, 0 );
#else
   tpreturn(TPSUCCESS,0,(char
*)bpd,sizeof(delivery_trans),0          );
#endif
   }
   else if( mix == 5 )
   {
#ifdef USE_FML                          /* 98.04.07
Ich. */
          sbuf = *( ( stocklvl_trans * )Ffind( ( FBFR
* )info->data,
             FML_DATA, 0, NULL ) );
          bps = &sbuf;
#else
      bps = (stocklvl_trans *)info->data;
#endif
      rtnsize = sizeof(stocklvl_trans);
      w_id = bps->w_id ;
      d_id = bps->d_id ;
      threshold = bps->threshold;


      if(StockLevel())
      {
         bps->C_R = 1;
         bps->low_stock = low_stock;
       }
       else
       {
         bps->C_R = 0;
       }
#ifdef USE_FML                          /* 98.04.07
Ich. */
```

```
        Fchg( ( FBFR * )info->data, FML_DATA, 0, ( char
* )bps,
            sizeof( stocklvl_trans ) );
#endif
    }

        if( mix != 4 )
        {
#ifdef  TRACE
    DP("tpreturn-called mix=%d \n",mix);
#endif
#ifdef USE_FML                                  /* 98.04.07
Ich. */
        tpreturn( TPSUCCESS, 0, info->data, 0L, 0 );
#else
        tpreturn(TPSUCCESS,0,info->data,rtnsize,0);
#endif
        }
}

/****************************************************/
/*  tpsvrdone                                       */
/****************************************************/
void tpsvrdone()
{
#ifdef  TRACE
        DP("tpsvrdone called pid=%d\n",getpid());
#endif
        JMPCINT3()              ;
        fflush(delivery_handle);
        fclose(delivery_handle);
#ifdef USE_SQL_MODE                             /* 98.02.23
suzuki */
        EXEC SQL COMMIT WORK    ;
#endif
            /*                          */
            EXEC SQL DISCONNECT CURRENT ;
            DP("DISCONNECT(SQLSTATE) = %s\n",
SQLSTATE) ;
        return;
}

/****************************************************/
/*  Error                                           */
/****************************************************/
int Error()
{
        char msg[1024];
        long errno;
        FILE *handle;
        SQLSTATE[5] = 0 ;
        if (0 != strcmp(SQLSTATE,"00000") )
         {
           if (0 == strcmp(SQLSTATE,"40001")) /*
*/
            {
              return(1);
            }
/*      98.03.02                */
/*                                              "tpccerr"
*/
#ifdef NT
                        system("date /T>>tpccerr");
#else
            system("date >>/tmp/tpccerr");
#endif

#ifdef NT
            handle = fopen("tpccerr","ab");
#else
            handle = fopen("/tmp/tpccerr","ab");
#endif
/*      98.03.02        */

            if ( handle == NULL )
             {
               handle = stderr;
             }
            fprintf(handle," SQL
ERROR:SQLSTATE= %s\n",SQLSTATE);
            fflush(handle);

        }
        return(0);
}

/****************************************************/
/*  tpsvrinit                                       */
/****************************************************/
tpsvrinit(argc,argv)
int argc;
char **argv;
{
        int i = 0;
        char *fname;

#ifdef suzuki
/*                  98.07.01      */
         if(tpsvrinit_fp == 0){

         sprintf(fname,"tpsvrinit_test%d.txt",getpid());
                 tpsvrinit_fp = fopen(fname,"w");
         }
         fprintf(tpsvrinit_fp,"tpsvrinit start\n");
         fflush(tpsvrinit_fp);
/* 98.07.01       */
#endif


        DP("tpsvrinit start  called pid=%d\n",getpid());



#ifdef USE_SQL_MODE                             /* 98.02.23
suzuki */
        EXEC SQL WHENEVER SQLERROR
CONTINUE;
#endif
         /*                                     */
#ifdef USE_SQL_MODE                             /* 98.02.23
suzuki */
            EXEC SQL CONNECT TO 'SV1';
/*DEFAULT;*/
#endif
            DP("CONNECT(SQLSTATE) = %s\n",
SQLSTATE) ;

#ifdef USE_SQL_MODE                             /* 98.02.23
suzuki */
        EXEC SQL COMMIT WORK ;
#endif
        JMPCINT2();

        if(! preNewOrder() ) ++i;
        if(! prePayment() ) ++i;
        if(! preOrderStatus() ) ++i;
        if(! preDelivery() ) ++i;
        if(! preStockLevel() ) ++i;

        if(i)
        {
                printf("%d errors in SQL
prepares.quitting.\n",i);
                fflush(stdout);
                exit(1);
        }


        DP("tpsvrinit end  called pid=%d\n",getpid());



#ifdef suzuki
/*                  98.07.01      */
         if(tpsvrinit_fp == 0){
                 tpsvrinit_fp = fopen("tpsvrinit_test.txt","w");
         }
         fprintf(tpsvrinit_fp,"tpsvrinit end\n");
         fflush(tpsvrinit_fp);
/* 98.07.01       */
#endif

}

int scanstring(target,search,length)
char *target,*search;
int length;
{
        int search_length,iter;
        if((search_length = strlen(search)) > length)
        {
                return(-1);
        }
        for (iter= length -search_length;iter;--iter,++target)
        {
                if(strncmp(target,search,search_length) == 0)
                {
                        return(1);
                }
        }
        return(0);
}

/****************************************************/
/*  preNewOrder                                     */
/****************************************************/
preNewOrder()
{
    return(1);
}

/****************************************************/
/*  NewOrder                                        */
/****************************************************/
NewOrder()
{
    long i_price[15];
    char i_name[15][25];
    char i_data[15][51];
    char s_datax[15][51];
        lnk_ol lnk_buf[15];                     /*    INSERT
*/
    int  j  ;
    int  i  ;
```

```
    long total_amount = 0;
    int  pos = 0;
    int  in_ol_i_id             ;
    int  in_ol_number           ;
         s_join_str     *sjp    ;
         i_join_str     *ijp    ;
         ol_i_join_str  *olijp  ;
         ol_s_join_str  *olsjp  ;
         ol_q_join_str  *olqjp  ;
    int  item_notfound_cnt      ;
    FILE *fp;
    FILE *handle;
    int  retry_flag = 0 ;

short     *ol_i_id_ind_str[]   = { (short *)&ol_i_id1_ind    ,
                                   (short *)&ol_i_id2_ind    ,
                                   (short *)&ol_i_id3_ind    ,
                                   (short *)&ol_i_id4_ind    ,
                                   (short *)&ol_i_id5_ind    ,
                                   (short *)&ol_i_id6_ind    ,
                                   (short *)&ol_i_id7_ind    ,
                                   (short *)&ol_i_id8_ind    ,
                                   (short *)&ol_i_id9_ind    ,
                                   (short *)&ol_i_id10_ind   ,
                                   (short *)&ol_i_id11_ind   ,
                                   (short *)&ol_i_id12_ind   ,
                                   (short *)&ol_i_id13_ind   ,
                                   (short *)&ol_i_id14_ind   ,
                                   (short *)&ol_i_id15_ind   ,
                                   NULL};
 struct {
         int    num          ;
         long   ol_i_id      ;
       } sort_id[15]         ;
 struct {
         int    num          ;
         long   ol_i_id      ;
       } r_id[15]            ;
         int    sort_num     ;
         long   sort_ol_i_id ;
#ifdef USE_SQL_MODE                          /* 98.02.23
suzuki */
EXEC SQL BEGIN DECLARE SECTION;
#endif
         short  h_cnt        ;
         short  r_cnt        ;
#ifdef USE_SQL_MODE                          /* 98.02.23
suzuki */
EXEC SQL END   DECLARE SECTION;
#endif

begin_tran:
#ifdef USE_SQL_MODE                          /* 98.02.23
suzuki */
    EXEC SQL WHENEVER SQLERROR
GOTO :sqlerr    ;
    EXEC SQL WHENEVER NOT FOUND
GOTO :not_found ;
#endif

        errorpos         = 0                        ;
        item_notfound    = -1                       ;
        s_join.sqllen    = 0                        ;
        i_join.sqllen    = 0                        ;
        sjp              = (s_join_str *)&s_join    ;
        ijp              = (i_join_str *)&i_join    ;
        olijp            = (ol_i_join_str *)&ol_i_join ;
        olsjp            = (ol_s_join_str *)&ol_s_join ;
        olqjp            = (ol_q_join_str *)&ol_q_join ;

        h_cnt = 0 ;
        r_cnt = 0 ;
        for (ol_number = 0; ol_number <
o_ol_cnt ;++ol_number)
        {
          if ( w_id == bpn->ol_supply_w_id[ol_number] )
           {
             for ( i=0 ; i < h_cnt ; i++ )
              {
                if ( sort_id[i].ol_i_id == bpn-
>ol_i_id[ol_number] )
                  {
                    break ;
                  }
              }
             if ( i == h_cnt )
              {
                sort_id[h_cnt].num     =
ol_number               ;
                sort_id[h_cnt].ol_i_id = bpn-
>ol_i_id[ol_number]  ;
                h_cnt = h_cnt + 1                          ;
              }
             else
              {
                r_id[r_cnt].num        =
ol_number               ;
                r_id[r_cnt].ol_i_id    = bpn-
>ol_i_id[ol_number]  ;
                r_cnt = r_cnt + 1                          ;
              }
           }
          else
           {
                r_id[r_cnt].num        =
ol_number               ;
                r_id[r_cnt].ol_i_id    = bpn-
>ol_i_id[ol_number]  ;
                r_cnt = r_cnt + 1                          ;
           }
        }

retry_neworder:
        if ( ( h_cnt < 5 ) || ( retry_flag == 1 ) )
         {  /*                                          */
           h_cnt = 0 ;
           r_cnt = 0 ;
           for (ol_number = 0; ol_number <
o_ol_cnt ;++ol_number)
           {
                r_id[r_cnt].num        =
ol_number               ;
                r_id[r_cnt].ol_i_id    = bpn-
>ol_i_id[ol_number]  ;
                r_cnt = r_cnt + 1                          ;
           }
         }

        for ( i=0 ; i < r_cnt ; i++ )
        {
                sort_id[h_cnt+i].num     =
r_id[i].num             ;
                sort_id[h_cnt+i].ol_i_id =
r_id[i].ol_i_id         ;
        }

        for (ol_number = 0; ol_number <
h_cnt ;++ol_number)
        {
          for (in_ol_number = ol_number + 1  ;
               in_ol_number <
h_cnt          ;++in_ol_number)
           {
             if (sort_id[in_ol_number].ol_i_id >
sort_id[ol_number].ol_i_id)
               {
                sort_num                   =
sort_id[ol_number].num       ;
                sort_ol_i_id               =
sort_id[ol_number].ol_i_id   ;
                sort_id[ol_number].num     =
sort_id[in_ol_number].num    ;
                sort_id[ol_number].ol_i_id    =
sort_id[in_ol_number].ol_i_id;
                sort_id[in_ol_number].num     =
sort_num                     ;
                sort_id[in_ol_number].ol_i_id =
sort_ol_i_id                 ;
               }
           }
        }

        for (i=0,ol_number = 0; ol_number <
15 ;++ol_number)
        {
          if (ol_number < h_cnt)
           {
             *((short *)(ol_i_id_ind_str[ol_number])) =
0          ;
             *((long  *)(ol_i_id_str[ol_number]))
                  = bpn-
>ol_i_id[sort_id[ol_number].num]          ;

             sprintf(olqjp-
>sqlvar[ol_number].ol_quantity,"%-4d",
                     bpn-
>ol_quantity[sort_id[ol_number].num] )    ;
           }
          else
           {
             *((short *)(ol_i_id_ind_str[ol_number])) = -
1          ;
             *((long  *)(ol_i_id_str[ol_number]))     =
0          ;

             if ( ol_number < o_ol_cnt )
              {
                sprintf(olqjp-
>sqlvar[ol_number].ol_quantity,"%-4d",
                     bpn-
>ol_quantity[sort_id[ol_number].num] )    ;

                sprintf(olijp->sqlvar[i].ol_i_id,"%-7d",
                     bpn-
>ol_i_id[sort_id[ol_number].num] )        ;
                sprintf(olsjp->sqlvar[i].ol_supply_w_id,"%-
4d",
                     bpn-
>ol_supply_w_id[sort_id[ol_number].num] ) ;
                i++ ;
              }
           }
        }
        ol_q_join.sqllen = o_ol_cnt * 4                ;
```

```
        ol_i_join.sqllen = r_cnt    * 7              ;
        ol_s_join.sqllen = r_cnt    * 4              ;

        s_ymdhms()                                   ;
        strncpy(o_entry_d, tc_s,14)                  ;
        bpn->o_entry_d = t_wk                        ;

neworder_proc:
#ifdef USE_SQL_MODE                          /* 98.02.23
suzuki */
          EXEC SQL
            CALL
TPCC_SCHEMA.Y_NORDER5(:state                         ,
                :errorpos
INDICATOR :errorpos_ind      ,
                :w_id                                ,
                :tmp_d_id                            ,
                :c_id                                ,
                :o_all_local
INDICATOR :o_all_local_ind   ,
                :w_tax
INDICATOR :w_tax_ind         ,
                :d_tax
INDICATOR :d_tax_ind         ,
                :o_id
INDICATOR :o_id_ind          ,
                :o_entry_d                           ,
                :c_discount
INDICATOR :c_discount_ind    ,
                :c_last
INDICATOR :c_last_ind        ,
                :c_credit
INDICATOR :c_credit_ind      ,
                :item_notfound
INDICATOR :item_notfound_ind ,
                :h_cnt                               ,
                :r_cnt                               ,
                  :ol_i_id1                          ,
                  :ol_i_id2                          ,
                  :ol_i_id3                          ,
                  :ol_i_id4                          ,
                  :ol_i_id5                          ,
                  :ol_i_id6
INDICATOR :ol_i_id6_ind      ,
                  :ol_i_id7
INDICATOR :ol_i_id7_ind      ,
                  :ol_i_id8
INDICATOR :ol_i_id8_ind      ,
                  :ol_i_id9
INDICATOR :ol_i_id9_ind      ,
                  :ol_i_id10
INDICATOR :ol_i_id10_ind     ,
                  :ol_i_id11
INDICATOR :ol_i_id11_ind     ,
                  :ol_i_id12
INDICATOR :ol_i_id12_ind     ,
                  :ol_i_id13
INDICATOR :ol_i_id13_ind     ,
                  :ol_i_id14
INDICATOR :ol_i_id14_ind     ,
                  :ol_i_id15
INDICATOR :ol_i_id15_ind     ,
                  :ol_i_join                         ,
                  :ol_q_join                         ,
                  :s_join                            ,
                  :i_join                            ,
                  :ol_s_join
                        );
#else
                SQLWAIT_N;
        strcpy(state,"00000");

//      o_id = rand()%99999999+1;           /* 98.03.24
Ich. */
        o_id = 3001;                        /* 98.10.14
Moriai */
        bpn->w_tax = rand()%2001;
        bpn->d_tax = rand()%2001;
        strcpy( bpn->c_last, "BAROUGHTABLE" );
        strcpy( bpn->c_credit, "GC" );
        bpn->c_discount = rand()%101;
#endif

        if ( memcmp(state,"00000",5) != 0 )
         {
           strncpy(SQLSTATE,state,5) ;
           SQLSTATE[5] = 0          ;
           if ( memcmp(state,"02000",5) == 0 )
            {
              goto not_found;
            }
           else
            {
              goto sqlerr;
            }
         }

        if ( item_notfound == -2 )
         {  /*                                  */
           retry_flag = 1 ;
           goto retry_neworder ;
         }

#ifdef  DP_IJ
                DP("IJ: item_notfound=%d \n",
item_notfound);
#endif
        bpn->o_id       = o_id          ;

        for ( ol_number = 0;ol_number <
o_ol_cnt;++ol_number )
        {
          ol_i_id = bpn->ol_i_id[ol_number] ;
          /*      99.05.27
*/
          for ( in_ol_number = 0;in_ol_number <
o_ol_cnt;++in_ol_number )
          {
            if ( ol_number == sort_id[in_ol_number].num )
             {
               i_price[ol_number]
                        = atoi(ijp-
>sqlvar[in_ol_number].i_price)   ;
                if ( i_price[ol_number] == 0 )
                 {
                   /*                           */
                   bpn->i_price[ol_number]    = 0          ;
                   bpn->s_quantity[ol_number] =
0           ;
                   ol_dist_info[0]          = '\0'         ;
                   bpn->i_name[ol_number][0]  = '\0'       ;
                   break ;
                 }
                else
                 {
                   bpn->i_price[ol_number] =
i_price[ol_number]          ;

                   strncpy(bpn->i_name[ol_number],
                           ijp-
>sqlvar[in_ol_number].i_name,24)      ;
                   bpn->i_name[ol_number][24] =
'\0'                ;

                   strncpy(i_data[ol_number],
                           ijp-
>sqlvar[in_ol_number].i_data,50)      ;
                   i_data[ol_number][50]  =
'\0'                ;
                   bpn->s_quantity[ol_number]
                           = atoi(sjp-
>sqlvar[in_ol_number].s_quantity) ;
                   strncpy(ol_dist_info,
                           sjp-
>sqlvar[in_ol_number].s_dist,24)      ;
                   ol_dist_info[24] =
'\0'                ;
                   strncpy(s_datax[ol_number],
                           sjp-
>sqlvar[in_ol_number].s_data,50)      ;
                   s_datax[ol_number][50]       =
'\0'                ;
                   /*sort_id[in_ol_number].ol_i_id = 0 ;
1997.02.24 */
                   break ;
                 }
#ifdef  DP_IJ
                DP("IJ: ol_num=%d,", ol_number);
                DP("    price =%d,", i_price[ol_number]);
                DP("    name  =%s \n", &(bpn-
>i_name[ol_number][0]) );
#endif
            }
          }
         /*     99.05.27     */

         ol_amount              = bpn-
>ol_quantity[ol_number]
                                * i_price[ol_number]        ;
         bpn->ol_amount[ol_number] =
ol_amount                   ;
         total_amount           += ol_amount                ;
         if ( scanstring(i_data[ol_number],"ORIGINAL",50)
          &&
scanstring(s_datax[ol_number],"ORIGINAL",50) )
           {
             bpn->brand_generic[ol_number] = 'B';
           }
         else
           {
             bpn->brand_generic[ol_number] = 'G';
           }

         /***     INSERT ***/
         lnk_buf[ol_number].ol_o_id       =
o_id                ;
         lnk_buf[ol_number].ol_d_id       =
tmp_d_id            ;
         lnk_buf[ol_number].ol_w_id       =
w_id                ;
         lnk_buf[ol_number].ol_number     = ol_number +
1       ;
```

```
        lnk_buf[ol_number].ol_i_id       =
ol_i_id             ;
        lnk_buf[ol_number].ol_supply_w_id
                          = bpn-
>ol_supply_w_id[ol_number] ;
        lnk_buf[ol_number].ol_quantity
                          = bpn-
>ol_quantity[ol_number]    ;
        lnk_buf[ol_number].ol_amount     =
ol_amount           ;
        strncpy(lnk_buf[ol_number].ol_dist_info,ol_dist_i
nfo,25) ;
       }

#ifdef USE_SQL_MODE                          /* 98.02.23
suzuki */
     /*---- ORDERLINE INSERT ----*/
     errorpos = 108 ;
     j =
OLINSERT(&lnk_buf[0],&o_ol_cnt,&SQLSTATE);
     if ( j != 0)
      {
        DP(" NewOrder ERRPOS=%d
SQLSTATE=%s\n",errorpos, SQLSTATE );
        goto sqlerr;
      }

     if ( item_notfound == -1)
      {
        EXEC SQL COMMIT WORK                         ;

        strncpy(bpn->c_last,c_last,17)               ;
        strncpy(bpn->c_credit,c_credit,3)            ;
        bpn->d_tax      = d_tax                      ;
        bpn->w_tax      = w_tax                      ;
        bpn->c_discount = c_discount                 ;
        total_amount *= (1 + (w_tax + d_tax)/10000.0)
                  * (1 - (c_discount    /10000.0)) ;
        bpn->total_amount = total_amount             ;
        bpn->errorpos     = 0                        ;
        bpn->sqlstate     = 0                        ;
        return(1)                                    ;
       }
     else
      {
        /*     99.05.27              c_last,c_credit
*/
        strncpy(bpn->c_last,c_last,17)               ;
        strncpy(bpn->c_credit,c_credit,3)            ;
        /*     99.05.27      */
        bpn->errorpos     = 201                      ;
        bpn->sqlstate     = 02000                    ;

        EXEC SQL ROLLBACK
WORK                         ;
        return(2)                                    ;
       }
#else
            SQLWAIT_N_C;
            SQLWAIT_N_R;

      bpn->total_amount = 0;                 /* 98.03.24
Ich. */
     for ( i = 0; i < 15; ++i )
      {
        if ( bpn->ol_supply_w_id[i] == 0 ) {
          break;
        }
        strcpy( bpn->i_name[i],
"NAMESNAMESNAMESNAME" );
        bpn->s_quantity[i] = ( rand()%10 ) + 1;
        bpn->brand_generic[i] = 'G';
        bpn->i_price[i] = ( rand()%9901 )+100;
        bpn->ol_amount[i] = bpn->i_price[i]*bpn-
>ol_quantity[i];
        bpn->total_amount += bpn->ol_amount[i];
       }
       bpn->o_ol_cnt = i;
      /* bpn->total_amount *= ( 1.0 + ( bpn->w_tax +
bpn->d_tax )/10000.0 )
                    * ( 1.0 - ( bpn-
>c_discount/10000.0 ) ); */
            return(1)                                ;
#endif

not_found:
   DP("NOT FOUND IN NewOrder AT %d\n",errorpos);
   fflush(stdout);
   bpn->errorpos     = errorpos               ;
   bpn->sqlstate     = atoi(SQLSTATE)         ;
#ifdef USE_SQL_MODE                          /* 98.02.23
suzuki */
   EXEC SQL ROLLBACK WORK;
#else
       SQLWAIT_N_R;
#endif
   return(0);

sqlerr:
#ifdef  DP_SQLERR
   DP("Neworder ERRPOS=%d
SQLSTATE=%s\n",errorpos,SQLSTATE);
#endif
#ifdef USE_SQL_MODE                          /* 98.02.23
suzuki */
   EXEC SQL WHENEVER SQLERROR  CONTINUE ;
#endif
   if(Error())
     {
#ifdef USE_SQL_MODE                          /* 98.02.23
suzuki */
            EXEC SQL ROLLBACK WORK;
#else
       SQLWAIT_N_R;
#endif
            goto begin_tran;
     }
   bpn->errorpos     = errorpos               ;
   bpn->sqlstate     = atoi(SQLSTATE)         ;
#ifdef USE_SQL_MODE                          /* 98.02.23
suzuki */
   EXEC SQL ROLLBACK WORK;
#else
       SQLWAIT_N_R;
#endif
   return(0);
}

/************************************************/
/*  prePayment                                  */
/************************************************/
prePayment()
{
   return(1);
}

/************************************************/
/*  Payment                                     */
/************************************************/
Payment()
{

begin_tran:
#ifdef USE_SQL_MODE                          /* 98.02.23
suzuki */
   EXEC SQL WHENEVER SQLERROR
GOTO :sqlerr    ;
   EXEC SQL WHENEVER NOT FOUND
GOTO :not_found ;
#endif

   s_ymdhms()                    ;
   strncpy(h_date, tc_s,14)      ;

   c_datax.sqllen     = 0;  /*99.05.11 c_data     */
   c_datax.sqlvar[0]  = 0;  /*99.05.11 c_data     */
   c_datax.sqlvar[500] = 0;  /*99.05.11 c_data     */

       errorpos = 0            ;
#ifdef USE_SQL_MODE                          /* 98.02.23
suzuki */
       EXEC SQL
         CALL
TPCC_SCHEMA.Y_PAYMENT_H10_OUT4(:state
,
                :errorpos
INDICATOR :errorpos_ind     ,
                :w_id                                ,
                :d_id                                ,
                :c_id                                ,
                :c_d_id                               ,
                :c_w_id                               ,
                :h_amount                              ,
                :h_date                               ,
                :w_name                                ,
                :w_street_1
INDICATOR :w_street_1_ind   ,
                :w_street_2
INDICATOR :w_street_2_ind   ,
                :w_city
INDICATOR :w_city_ind       ,
                :w_state
INDICATOR :w_state_ind      ,
                :w_zip
INDICATOR :w_zip_ind        ,
                :d_name                                ,
                :d_street_1
INDICATOR :d_street_1_ind   ,
                :d_street_2
INDICATOR :d_street_2_ind   ,
                :d_city
INDICATOR :d_city_ind       ,
                :d_state
INDICATOR :d_state_ind      ,
                :d_zip
INDICATOR :d_zip_ind        ,
                :c_first
INDICATOR :c_first_ind      ,
                :c_middle
INDICATOR :c_middle_ind     ,
                :c_last                              ,
                :c_street_1
INDICATOR :c_street_1_ind   ,
```

```
                :c_street_2
INDICATOR :c_street_2_ind    ,
                :c_city
INDICATOR :c_city_ind        ,
                :c_state
INDICATOR :c_state_ind       ,
                :c_zip
INDICATOR :c_zip_ind         ,
                :c_phone
INDICATOR :c_phone_ind       ,
                :c_credit                                ,
                :c_credit_lim
INDICATOR :c_credit_lim_ind  ,
                :c_discount
INDICATOR :c_discount_ind    ,
                :c_balance
INDICATOR :c_balance_ind     ,
                :c_ytd_payment
INDICATOR :c_ytd_payment_ind ,
                :c_payment_cnt
INDICATOR :c_payment_cnt_ind ,
                :c_since
INDICATOR :c_since_ind ,
                :c_datax            INDICATOR :c_data_ind
                        );
#else
              SQLWAIT_P;
        strcpy(state,"00000");

        c_discount = rand()%5001;               /* 98.03.24
Ich. */
        strcpy( c_first, "ABCDEFGHIJKL" );
        strcpy( c_middle, "OE" );
        strcpy( c_last, "BAROUGHTABLE" );
        strcpy( c_phone, "0123456789012345" );
        c_id = rand()%3000 + 1;

        strcpy( c_street_1, "PQRSTUVWXYZABCD" );
        strcpy( c_street_2, "EFGHIJKOLMNOPQR" );
        strcpy( c_city, "STUVWXYZABCDEFG" );
        strcpy( c_state, "RE" );
        sprintf( c_zip, "%04d11111", rand()%10000 );

        strcpy( d_street_1, "PQRSTUVWXYZABCD" );
        strcpy( d_street_2, "EFGHIJKOLMNOPQR" );
        strcpy( d_city, "STUVWXYZABCDEFG" );
        strcpy( d_state, "RE" );
        sprintf( d_zip, "%04d11111", rand()%10000 );

        strcpy( w_street_1, "PQRSTUVWXYZABCD" );
        strcpy( w_street_2, "EFGHIJKOLMNOPQR" );
        strcpy( w_city, "STUVWXYZABCDEFG" );
        strcpy( w_state, "RE" );
        sprintf( w_zip, "%04d11111", rand()%10000 );

        c_balance = ( ( rand()*rand()%19999999 )-
9999999 ) / 100.0;
        c_credit_lim = 5000000;

        strcpy( c_since, "19980212121212" );
        strcpy( c_credit, "GC" );
#endif

        if ( memcmp(state,"00000",5) != 0 )
          {
            strncpy(SQLSTATE,state,5) ;
            SQLSTATE[5] = 0          ;
            if ( memcmp(state,"02000",5) == 0 )
              {
                goto not_found;
              }
            else
              {
                goto sqlerr;
              }
          }

    bpp->c_discount = c_discount      ;
    bpp->h_date     = t_wk            ;
    strcpy(bpp->c_first,c_first)      ;
    strcpy(bpp->c_middle,c_middle)    ;
    strcpy(bpp->c_last,c_last)        ;
    strcpy(bpp->c_phone,c_phone)      ;
    bpp->c_id= c_id                   ;
    strcpy(bpp->c_street_1,c_street_1) ;
    strcpy(bpp->c_street_2,c_street_2) ;
    strcpy(bpp->c_city,c_city)        ;
    strcpy(bpp->c_state,c_state)      ;
    strcpy(bpp->c_zip,c_zip)          ;

    strcpy(bpp->d_street_1,d_street_1) ;
    strcpy(bpp->d_street_2,d_street_2) ;
    strcpy(bpp->d_city,d_city)        ;
    strcpy(bpp->d_state,d_state)      ;
    strcpy(bpp->d_zip,d_zip)          ;

    strcpy(bpp->w_street_1,w_street_1) ;
    strcpy(bpp->w_street_2,w_street_2) ;
    strcpy(bpp->w_city,w_city)        ;
    strcpy(bpp->w_state,w_state)      ;
    strcpy(bpp->w_zip,w_zip)          ;

//  bpp->c_balance = c_balance        ; 98.11.19 DB
    bpp->c_balance = c_balance / 100.0 ;
//  bpp->c_credit_lim = c_credit_lim  ; 98.11.19 DB
    bpp->c_credit_lim = c_credit_lim / 100.0 ;

    bpp->c_since = c_ymdhms(c_since)  ;
    strcpy(bpp->c_credit,c_credit)    ;

#ifdef USE_SQL_MODE                             /*
98.03.24 Ich. */
 if ( strcmp(c_credit,"BC") == 0 )
  {
    strncpy(bpp-
>c_data,c_datax.sqlvar,c_datax.sqllen);
  }
 else
  {
    bpp->c_data[0] = 0 ;
  }
#else                                           /* 98.03.24 Ich. */
  if ( rand()%10 == 1 ) {
#define _STR50
        "0123456789abcdefghijklmnopqrstuvwxyz!#$&'()=
-~[]:;"
    strcpy( bpp->c_credit, "BC" );
    strcpy( bpp->c_data,
      _STR50 _STR50 _STR50 _STR50 _STR50
_STR50 _STR50 _STR50 _STR50 );
  } else {
    bpp->c_data[0] = '\0';
  }
#endif

  bpp->errorpos     = 0 ;
  bpp->sqlstate     = 0 ;
  /*EXEC SQL COMMIT WORK;*/
  return(1);

not_found:
       DP("NOT FOUND IN Payment
AT %d\n",errorpos);
       fflush(stdout);
       bpp->errorpos     = errorpos              ;
       bpp->sqlstate     = atoi(SQLSTATE)        ;
       /*EXEC SQL ROLLBACK WORK;*/
       return(0);

sqlerr:
#ifdef  DP_SQLERR
    DP("Payment  ERRPOS=%d
SQLSTATE=%s\n",errorpos,SQLSTATE);
#endif
    if(Error())
    {
       /*EXEC SQL ROLLBACK WORK;*/
       goto begin_tran;
     }
    bpp->errorpos     = errorpos              ;
    bpp->sqlstate     = atoi(SQLSTATE)        ;
    /*EXEC SQL ROLLBACK WORK;*/
    return(0);
}


/************************************************/
/*  preOrderStatus                              */
/************************************************/
preOrderStatus()
{
    return(1);
}

/************************************************/
/*  OrderStatus                                 */
/************************************************/
OrderStatus()
{
       ol_join_str  *oljp                    ;

        /* 980821 add suzuki */
        int l;
        char ol_supply_w_id_5[5] = {0,0,0,0,0};
        /* 980821 suzuki */

begin_tran:
#ifdef USE_SQL_MODE                             /* 98.02.23
suzuki */
    EXEC SQL WHENEVER SQLERROR
GOTO :sqlerr    ;
    EXEC SQL WHENEVER NOT FOUND
GOTO :not_found ;
#endif

        ol_join.sqllen = 0                      ;
        oljp           = (ol_join_str *)&ol_join ;
        errorpos = 0        ;
        /*printf( "Order-status\n" );*/
#ifdef USE_SQL_MODE                             /* 98.02.23
suzuki */
        EXEC SQL
          CALL
TPCC_SCHEMA.Y_ODERSTAT(:state                   ,
```

```
                :errorpos
INDICATOR :errorpos_ind     ,
                :w_id                                   ,
                :d_id                                   ,
                :c_id                                   ,
                :c_first
INDICATOR :c_first_ind      ,
                :c_middle
INDICATOR :c_middle_ind     ,
                :c_last
INDICATOR :c_last_ind       ,
                :c_balance
INDICATOR :c_balance_ind    ,
                :o_id
INDICATOR :o_id_ind         ,
                :o_entry_d
INDICATOR :o_entry_d_ind    ,
                :o_carrier_id
INDICATOR :o_carrier_id_ind ,
                :o_ol_cnt                               ,
                :ol_join        INDICATOR :ol_join_ind
                        );
#else
            SQLWAIT_O;
      strcpy(state,"00000");

       c_id = rand()%3000 + 1;                  /* 98.03.24
Ich. */
       strcpy( c_first, "ABCDEFGHIJKL" );
       strcpy( c_middle, "OE" );
       strcpy( c_last, "BAROUGHTABLE" );
       c_balance = ( ( rand()*rand()%19999999 )-
9999999 ) / 100.0;
       o_id = rand()%99999999+1;
       strcpy( o_entry_d, "19980123123456" );
       o_ol_cnt = rand()%11 + 5;
#endif

      if ( memcmp(state,"00000",5) != 0 )
       {
         strncpy(SQLSTATE,state,5) ;
         SQLSTATE[5] = 0          ;
         if ( memcmp(state,"02000",5) == 0 )
          {
            goto not_found;
          }
         else
          {
            goto sqlerr;
          }
       }

#ifdef USE_SQL_MODE                             /*
98.03.24 Ich. */
      for ( ol_number = 0;ol_number <
o_ol_cnt;++ol_number )
      {
        bpo->ol_i_id[ol_number]   = atoi(oljp-
>sqlvar[ol_number].ol_i_id)  ;
        bpo->ol_amount[ol_number] = atoi(oljp-
>sqlvar[ol_number].ol_amount);

              /* 980821 adjust for over 1000WH suzuki */
                for(l = 0;l < 4;l++){
                     ol_supply_w_id_5[l] = oljp-
>sqlvar[ol_number].ol_supply_w_id[l];
                }
        bpo->ol_supply_w_id[ol_number]
                        = atoi(ol_supply_w_id_5)   ;
/*                      = atoi(oljp-
>sqlvar[ol_number].ol_supply_w_id)   ;*/
                /* 980821 suzuki */

        bpo->ol_quantity[ol_number]
                        = atoi(oljp-
>sqlvar[ol_number].ol_quantity)     ;
        if(memcmp(oljp-
>sqlvar[ol_number].ol_delivery_d,"777777777",9) != 0)
         {
           bpo->ol_delivery_d[ol_number]
                        = c_ymdhms(oljp-
>sqlvar[ol_number].ol_delivery_d);
         }
        else
         {
           bpo->ol_delivery_d[ol_number] =
777777777                       ;
         }
      }
#else                                           /* 98.03.24 Ich. */
       for ( ol_number = 0; ol_number < o_ol_cnt;
++ol_number )
       {
         bpo->ol_i_id[ol_number] = ( rand()%100000 )+1;
         bpo->ol_amount[ol_number] = rand()%1000000;
         bpo->ol_supply_w_id[ol_number] =
( rand()%10 )+1;
         bpo->ol_quantity[ol_number] = ( rand()%99 )+1;
         bpo->ol_delivery_d[ol_number] =
c_ymdhms( "19980321054321" );
       }
#endif

       if ( o_carrier_id_ind == -1 )
        {
          bpo->o_carrier_id = INTNULL        ;
        }
       else
        {
          bpo->o_carrier_id = o_carrier_id   ;
        }
       bpo->c_id        = c_id           ;
       bpo->o_ol_cnt    = o_ol_cnt       ;
       strcpy(bpo->c_first,c_first)       ;
       strcpy(bpo->c_middle,c_middle)     ;
       strcpy(bpo->c_last,c_last)         ;
//     bpo->c_balance   = c_balance      ; 98.11.19 DB
       bpo->c_balance   = c_balance/100.0 ;
       bpo->o_id        = o_id           ;
       bpo->o_entry_d   = c_ymdhms(o_entry_d) ;

       bpo->errorpos    = 0              ;
       bpo->sqlstate    = 0              ;
       /*EXEC SQL COMMIT WORK;*/

    return (1);

not_found:
       DP("NOT FOUND IN OrderStatus
AT %d\n",errorpos);
       fflush(stdout);
       bpo->errorpos    = errorpos          ;
       bpo->sqlstate    = atoi(SQLSTATE)    ;
       /*EXEC SQL ROLLBACK WORK;*/
       return(0);

sqlerr:
#ifdef  DP_SQLERR
    DP("OrderStatus ERRPOS=%d
SQLSTATE=%s\n",errorpos,SQLSTATE);
#endif
    if(Error())
    {
       /*EXEC SQL ROLLBACK WORK;*/
       goto begin_tran;
    }

    bpo->errorpos    = errorpos          ;
    bpo->sqlstate    = atoi(SQLSTATE)    ;
    /*EXEC SQL ROLLBACK WORK;*/

    return(0);
}

/************************************************/
/*  preDelivery                                 */
/************************************************/
preDelivery()
{
       return(1);
}

/************************************************/
/*  Delivery                                    */
/************************************************/
Delivery()
{
       int     temp_d_id                ;
               result_join_str  *rjp    ;

begin_tran:
#ifdef USE_SQL_MODE                             /* 98.02.23
suzuki */
       EXEC SQL WHENEVER SQLERROR
GOTO :sqlerr    ;
       EXEC SQL WHENEVER NOT FOUND
GOTO :not_found ;
#endif

         result_join.sqllen = 0                 ;
         rjp = (result_join_str *)&result_join  ;
         s_ymdhms()                             ;
         strncpy(ol_delivery_d, tc_s,14)        ;
         o_carrier_id  = bpd->o_carrier_id      ;
         errorpos      = 0                      ;
#ifdef USE_SQL_MODE                             /* 98.02.23
suzuki */
         EXEC SQL
            CALL
TPCC_SCHEMA.Y_DELIVERY(:state                   ,
                :errorpos
INDICATOR :errorpos_ind     ,
                :w_id                                   ,
                :c_id                                   ,
                :o_carrier_id                           ,
                :ol_delivery_d                          ,
                :result_join
INDICATOR :result_join_ind
                        );
#else
              SQLWAIT_D;
       strcpy(state,"00000");
#endif
```

```
        if ( memcmp(state,"00000",5) != 0 )
         {
           strncpy(SQLSTATE,state,5) ;
           SQLSTATE[5] = 0          ;
           if ( memcmp(state,"02000",5) == 0 )
            {
              goto not_found;
            }
           else
            {
              goto sqlerr;
            }
         }

        for ( temp_d_id = 0 ; temp_d_id < 10 ;
temp_d_id++ )
         {
           result_o_id[temp_d_id] = atoi(rjp-
>sqlvar[temp_d_id].result_o_id);
         }

        bpd->errorpos     = 0 ;
        bpd->sqlstate     = 0 ;
        /*EXEC SQL COMMIT WORK;*/

        return(1);

not_found:
        DP("NOT FOUND IN DELIVERY
AT %d\n",errorpos);
        fflush(stdout);
        bpd->errorpos     = errorpos              ;
        bpd->sqlstate     = atoi(SQLSTATE)        ;
        /*EXEC SQL ROLLBACK WORK;*/
        return(0);

sqlerr:
#ifdef  DP_SQLERR
    DP("Delivery ERRPOS=%d
SQLSTATE=%s\n",errorpos,SQLSTATE);
#endif
        if(Error())
        {
              /*EXEC SQL ROLLBACK WORK;*/
              goto begin_tran;
        }
        bpd->errorpos     = errorpos              ;
        bpd->sqlstate     = atoi(SQLSTATE)        ;
        /*EXEC SQL ROLLBACK WORK;*/
        return(0);
}

/************************************************/
/*  preStockLevel                               */
/************************************************/
preStockLevel()
{
        return(1);
}

/************************************************/
/*  StockLevel                                  */
/************************************************/
StockLevel()
{

begin_tran:
#ifdef USE_SQL_MODE                     /* 98.02.23
suzuki */
        EXEC SQL WHENEVER SQLERROR
GOTO :sqlerr    ;
        EXEC SQL WHENEVER NOT FOUND
GOTO :not_found ;
#endif
          errorpos = 0                          ;

/* 98.06.12                                     */
#ifdef STOCK_STORED
#ifdef USE_SQL_MODE                     /* 98.02.23
suzuki */
          EXEC SQL
            CALL
TPCC_SCHEMA.Y_STOCKLV(:state                ,
                    :errorpos
INDICATOR :errorpos_ind ,
                    :w_id                   ,
                    :d_id                   ,
                    :threshold              ,
                    :low_stock
INDICATOR :low_stock_ind
                                  );
#else
        SQLWAIT_S;
        strcpy(state,"00000");

        low_stock = rand()%201;                 /*
98.03.24 Ich. */
#endif

        if ( memcmp(state,"00000",5) != 0 )
         {
           strncpy(SQLSTATE,state,5) ;
           SQLSTATE[5] = 0          ;
           if ( memcmp(state,"02000",5) == 0 )
            {
              goto not_found;
            }
           else
            {
              goto sqlerr;
            }
         }
#else
/* (1) DISTRICT   teble select */
                    EXEC SQL WHENEVER
SQLERROR  GOTO :ERR_S_DI;
                    EXEC SQL WHENEVER NOT
FOUND GOTO :ERR_S_DI;
        EXEC SQL SELECT D_NEXT_O_ID
             INTO :o_id
             FROM TPCC_SCHEMA.DISTRICT
             WHERE D_W_ID = :w_id
               AND D_ID   = :d_id;
                    EXEC SQL WHENEVER
SQLERROR  CONTINUE;
                    EXEC SQL WHENEVER NOT
FOUND CONTINUE;

/* (2) ORDERLINE  teble select */
/* (3) STOCK      teble select and count ITEM */

        tmp_o_id = o_id - 20;
        o_id = o_id - 1 ;
        t19  = o_id - 1;
        t18  = o_id - 2;
        t17  = o_id - 3;
        t16  = o_id - 4;
        t15  = o_id - 5;
        t14  = o_id - 6;
        t13  = o_id - 7;
        t12  = o_id - 8;
        t11  = o_id - 9;
        t10  = o_id - 10;
        t09  = o_id - 11;
        t08  = o_id - 12;
        t07  = o_id - 13;
        t06  = o_id - 14;
        t05  = o_id - 15;
        t04  = o_id - 16;
        t03  = o_id - 17;
        t02  = o_id - 18;

                    EXEC SQL WHENEVER
SQLERROR  GOTO :ERR_S_STOL;
                    EXEC SQL WHENEVER NOT
FOUND GOTO :ERR_S_STOL;

  EXEC SQL SELECT COUNT(DISTINCT S_I_ID)
           INTO  :low_stock
           FROM  TPCC_SCHEMA.ORDERLINE,
                 TPCC_SCHEMA.STOCK
           WHERE OL_W_ID   = :w_id
             AND OL_D_ID   = :d_id
             AND OL_O_ID
                IN(:tmp_o_id,
                   :t02,:t03,:t04,:t05,:t06,:t07,:t08,:t09,:
t10,
                   :t11,:t12,:t13,:t14,:t15,:t16,:t17,:t18,:t19,
                   :o_id )
             AND OL_NUMBER
IN(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)
--$
--$            AND OL_O_ID
--$                    BETWEEN @TMP_O_ID
--$                      AND @O_ID
             AND S_I_ID    = OL_I_ID
             AND S_W_ID    = OL_W_ID
             AND S_W_ID    = :w_id
             AND S_QUANTITY < :threshold;

                    EXEC SQL WHENEVER
SQLERROR  CONTINUE;
                    EXEC SQL WHENEVER NOT
FOUND CONTINUE;

/* 98.08.27 stocklevel      commit          */
           EXEC SQL COMMIT WORK;
/* 98.08.27     */

#endif
/* 98.06.12     */

        bps->errorpos     = 0 ;
        bps->sqlstate     = 0 ;
        /*EXEC SQL COMMIT WORK;*/
        return(1);

/* 98.06.12                                     */
/* --SQLERR:NOT_FOUND */
ERR_S_DI:
        bps->errorpos  = 203;
             goto sqlerr;
```

```
ERR_S_STOL:
        bps->errorpos  = 248;
                goto sqlerr;
/* 98.06.12     */


not_found:
        DP("NOT FOUND IN STOCKLEVEL
AT %d\n",errorpos);
        fflush(stdout);
        bps->errorpos     = errorpos              ;
        bps->sqlstate     = atoi(SQLSTATE)        ;
/* 98.08.27 stocklevel       rollback        (             )     */
        EXEC SQL ROLLBACK WORK;
/* 98.08.27     */
        return(0);

sqlerr:
#ifdef  DP_SQLERR
   DP("Stocklevel ERRPOS=%d
SQLSTATE=%s\n",errorpos,SQLSTATE);
#endif
        if(Error())
        {
/* 98.08.27 stocklevel       rollback        (             )     */
                EXEC SQL ROLLBACK WORK;
/* 98.08.27     */
                goto begin_tran;
        }
        bps->errorpos     = errorpos              ;
        bps->sqlstate     = atoi(SQLSTATE)        ;
/* 98.08.27 stocklevel       rollback        (             )     */
        EXEC SQL ROLLBACK WORK;
/* 98.08.27     */
        return(0);
}

#ifndef         USE_FML                                 /*
98.04.07 */
/************************************************/
/*  TPCC1->TPCC10 TPCCs1->3 TPCCd1->3          */
/************************************************/
TPCC1(info)
TPSVCINFO       *info;
        {
#ifdef  TRACE
        DP("TPCC-1 called\n");
#endif
        number = 1;
        return(TPCC(info,number));
        }
TPCC2(info)
TPSVCINFO       *info;
        {
        number = 2;
        return(TPCC(info,number));
        }
TPCC3(info)
TPSVCINFO       *info;
        {
        number = 3;
        return(TPCC(info,number));
        }
TPCC4(info)
TPSVCINFO       *info;
        {
        number = 4;
        return(TPCC(info,number));
        }
TPCC5(info)
TPSVCINFO       *info;
        {
        number = 5;
        return(TPCC(info,number));
        }
TPCC6(info)
TPSVCINFO       *info;
        {
        number = 6;
        return(TPCC(info,number));
        }
TPCC7(info)
TPSVCINFO       *info;
        {
        number = 7;
        return(TPCC(info,number));
        }
TPCC8(info)
TPSVCINFO       *info;
        {
        number = 8;
        return(TPCC(info,number));
        }
TPCC9(info)
TPSVCINFO       *info;
        {
        number = 9;
        return(TPCC(info,number));
        }
TPCC10(info)
TPSVCINFO       *info;
        {
        number = 10;
        return(TPCC(info,number));
        }
TPCCd1(info)
TPSVCINFO       *info;
        {
        number = 11;
        return(TPCC(info,number));
        }
TPCCd2(info)
TPSVCINFO       *info;
        {
        number = 12;
        return(TPCC(info,number));
        }
TPCCd3(info)
TPSVCINFO       *info;
        {
        number = 13;
        return(TPCC(info,number));
        }
TPCCs1(info)
TPSVCINFO       *info;
        {
        number = 14;
        return(TPCC(info,number));
        }
TPCCs2(info)
TPSVCINFO       *info;
        {
        number = 15;
        return(TPCC(info,number));
        }
TPCCs3(info)
TPSVCINFO       *info;
        {
        number = 16;
        return(TPCC(info,number));
        }
#endif
```

File: Y_DELIVERY

```
-- /**********STORED
PROCEDURE*********************************************/
-- /** Y_DELIVERY   COPYRIGHT FUJITSU LIMITED
1997                    **/
-- /**     :                                               **/
-- /**     :                                               **/
-- /**     :     SymfoWARE RDB TPC-C Benchmark
**/
-- /**     :    Delivery                                    **/
-- /**     :     1996/10/12                                  **/
-- /**          1997/03/13  Revision3.3 : Any Error(Clause
2.3.6)     **/
--
/*****************************************************************
********/

--  #RESULT_JOIN                VARCHAR(100)
--  +-------------------------------------+
--  | sqllen        short                 |
--  +-sqlver------------------------------+  ---
--  | #RESULT_O_IDn      CHAR(9)          |  |
--  +-------------------------------------+  |
--  |                    CHAR(1) "/"      |  |
--  +-------------------------------------+  ---
--
--  +-------------------------------------+
--
--
--
*****************************************************************
*************
EXEC SQL
CREATE PROCEDURE
TPCC_SCHEMA.Y_DELIVERY(OUT   #STATE
CHAR(5),
                                INOUT #ERRPOS
INTEGER  ,
                                IN    #W_ID
SMALLINT ,
                                INOUT #C_ID
INTEGER  ,
                                IN    #O_CARRIER_ID
SMALLINT ,
                                IN    #OL_DELIVERY_D
CHAR(14) ,
                                INOUT #RESULT_JOIN
VARCHAR(100)
                               )

DELIVERY:BEGIN
-- DECLARE

        DECLARE SQLSTATE            CHAR(5)
DEFAULT '00000';
        DECLARE SAPSTOP             CHAR(1)
DEFAULT '/'    ;
        DECLARE @OL_TOTAL           INTEGER ;
        DECLARE @DMY_W_ID           SMALLINT;
```

```
      DECLARE @DMY_D_ID          SMALLINT;
      DECLARE @D_ID              SMALLINT;
      DECLARE @NO_O_ID           INTEGER ;
--      DECLARE @OZAWK           SMALLINT;



-- (3) ORDERS    table cursor
      DECLARE CDOS CURSOR FOR
            SELECT O_C_ID
                FROM  TPCC_SCHEMA.ORDERS
                WHERE O_W_ID = #W_ID
                 AND O_D_ID = @D_ID
                 AND O_ID   = @NO_O_ID
             FOR UPDATE;

--   SET  @OZAWK = 1;

-- LOOP
  SET  @D_ID = 1;
  DID10:LOOP
      IF  @D_ID > 10 THEN
         GOTO NORMAL_END ;
      END IF;
      -- (1) NEWORDER                          NO_O_ID
      --
                           WHENEVER SQLERROR
GOTO ERR_S_NO;
      SELECT MIN( NO_O_ID )
           INTO  @NO_O_ID
           FROM  TPCC_SCHEMA.NEWORDER
           WHERE NO_W_ID = #W_ID
            AND NO_D_ID = @D_ID;
                           WHENEVER SQLERROR
CONTINUE;
      IF   SQLSTATE <> '00000'
       OR @NO_O_ID IS NULL    THEN
         SET   @NO_O_ID  = 99999999    ;
         GOTO  NEXT_DID ;
      END IF;

      -- (2) NEW-ORDER            @NO_O_ID
      --
                           WHENEVER SQLERROR
GOTO ERR_D_NO;
                           WHENEVER NOT FOUND
GOTO ERR_D_NO;
      DELETE FROM TPCC_SCHEMA.NEWORDER
           WHERE NO_W_ID = #W_ID
            AND NO_D_ID = @D_ID
            AND NO_O_ID = @NO_O_ID ;

      -- (5) ORDER-LINE           OL_AMOUNT
                           WHENEVER SQLERROR
GOTO ERR_S_OL;
                           WHENEVER NOT FOUND
GOTO ERR_S_OL;
      SELECT SUM(OL_AMOUNT)
           INTO  @OL_TOTAL
           FROM  TPCC_SCHEMA.ORDERLINE
              WHERE OL_W_ID = #W_ID
                AND OL_D_ID = @D_ID
                AND OL_O_ID = @NO_O_ID
                AND OL_NUMBER
IN(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15);

--$$                 AND OL_NUMBER = @OZAWK;
--$ -- if OL index exist
--$            WHERE OL_W_ID = #W_ID
--$              AND OL_D_ID = @D_ID
--$              AND OL_O_ID = @NO_O_ID;

         -- ORDER-LINE
                           WHENEVER SQLERROR
GOTO ERR_U_OL;
                           WHENEVER NOT FOUND
GOTO ERR_U_OL;
      UPDATE      TPCC_SCHEMA.ORDERLINE
           SET   OL_DELIVERY_D =
#OL_DELIVERY_D
            WHERE    OL_W_ID = #W_ID
                AND OL_D_ID = @D_ID
                AND OL_O_ID = @NO_O_ID
                AND OL_NUMBER
IN(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15);

--$ -- if OL index exist
--$            WHERE OL_W_ID = #W_ID
--$              AND OL_D_ID = @D_ID
--$              AND OL_O_ID = @NO_O_ID;

      -- (3) ORDER             @NO_O_ID
      --
                           WHENEVER SQLERROR
GOTO ERR_S_OR;
                           WHENEVER NOT FOUND
GOTO ERR_S_OR;
      OPEN   CDOS;
      FETCH  CDOS INTO #C_ID;
                           WHENEVER SQLERROR
CONTINUE;
                           WHENEVER NOT FOUND
CONTINUE;

      -- (4) ORDER
                           WHENEVER SQLERROR
GOTO ERR_U_OR;
      UPDATE      TPCC_SCHEMA.ORDERS
           SET   O_CARRIER_ID = #O_CARRIER_ID
           WHERE CURRENT OF CDOS;
                           WHENEVER SQLERROR
CONTINUE;
      CLOSE  CDOS;

      -- (6)                      Customer
                           WHENEVER SQLERROR
GOTO ERR_U_CM;
                           WHENEVER NOT FOUND
GOTO ERR_U_CM;
      UPDATE      TPCC_SCHEMA.CUSTOMER
           SET   C_BALANCE      = C_BALANCE +
@OL_TOTAL,
                 C_DELIVERY_CNT =
C_DELIVERY_CNT + 1
           WHERE C_W_ID  = #W_ID
             AND C_D_ID  = @D_ID
             AND C_ID    = #C_ID;
                           WHENEVER SQLERROR
CONTINUE;
                           WHENEVER NOT FOUND
CONTINUE;
NEXT_DID:

      SET  #RESULT_JOIN = #RESULT_JOIN
                  || CAST(@NO_O_ID AS CHAR(9)) ||
SAPSTOP  ;
      SET  @D_ID = @D_ID + 1;

      COMMIT WORK        ;

  END LOOP DID10;
-- LOOP END

NORMAL_END:
      SET #STATE = '00000' ;
      LEAVE DELIVERY      ;

--SQLERR:NOT_OUND:
ERR_S_OR:
      SET #ERRPOS = 207    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;
      LEAVE DELIVERY       ;
ERR_S_OL:
      SET #ERRPOS = 208    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;
      LEAVE DELIVERY       ;
ERR_S_NO:
      SET #ERRPOS = 209    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;
      LEAVE DELIVERY       ;
ERR_U_CM:
      SET #ERRPOS = 305    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;
      LEAVE DELIVERY       ;
ERR_U_OR:
      SET #ERRPOS = 307    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;
      LEAVE DELIVERY       ;
ERR_U_OL:
      SET #ERRPOS = 308    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;
      LEAVE DELIVERY       ;
ERR_D_NO:
      SET #ERRPOS = 409    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;

END DELIVERY
END-EXEC;
```

File: Y_NORDER5

```
-- /**********STORED
PROCEDURE*****************************************/
-- /** Y_NORDER     COPYRIGHT FUJITSU LIMITED
1997                **/
-- /**    :                                      **/
-- /**    :                                      **/
-- /**    :    SymfoWARE RDB TPC-C Benchmark
**/
-- /**    :   NewOrder                           **/
-- /**    :    1996/10/12                         **/
-- /**         1997/03/13  Revision3.3 : Any Error(Clause
2.3.6)    **/
-- /**         1999/05/27                S_JOIN,I_JOIN
**/
```

```
--
/*****************************************************************
********/

--
*****************************************************************
************
-- #S_JOIN                  VARCHAR(1215)
-- +----------------------------------------+
-- | sqllen        short                    |
-- +-sqlver---------------------------------+  ---
-- | S_QUANTITYn     CHAR(6)          |  |
-- +----------------------------------------+  |
-- | S_DISTn         CHAR(24)         |  |
-- +----------------------------------------+  |
-- | S_DATAn         CHAR(50)         |  |
-- +----------------------------------------+  |
-- |                 CHAR(1) "/"      |  |
-- +----------------------------------------+  ---
-- |                                    |
--
-- +----------------------------------------+
--
--
--
-- #I_JOIN                  VARCHAR(1215)
-- +----------------------------------------+
-- | sqllen        short                    |
-- +-sqlver---------------------------------+  ---
-- | I_PRICEHn       CHAR(6)          |  |
-- +----------------------------------------+  |
-- | I_NAMEn         CHAR(24)         |  |
-- +----------------------------------------+  |
-- | I_DATAn         CHAR(50)         |  |
-- +----------------------------------------+  |
-- |                 CHAR(1) "/"      |  |
-- +----------------------------------------+  ---
-- |                                    |
--
-- +----------------------------------------+
--
--
--
*****************************************************************
************
EXEC SQL
CREATE PROCEDURE
TPCC_SCHEMA.Y_NORDER5(OUT   #STATE
CHAR(5),
                      INOUT #ERRPOS
INTEGER ,
                      IN    #W_ID
SMALLINT,
                      IN    #D_ID          SMALLINT,
                      IN    #C_ID          INTEGER ,
                      INOUT #O_ALL_LOCAL
SMALLINT,
                      OUT   #W_TAX
SMALLINT,
                      OUT   #D_TAX
SMALLINT,
                      INOUT #O_ID
INTEGER ,
                      IN    #O_ENTRY_D
CHAR(14),
                      OUT   #C_DISCOUNT
SMALLINT,
                      OUT   #C_LAST
CHAR(16),
                      OUT   #C_CREDIT
CHAR(2),
                      INOUT #ITEM_NF_CTR
SMALLINT,
                      IN    #H_CNT
SMALLINT,
                      IN    #R_CNT
SMALLINT,
                        IN    #OL_I_ID1
INTEGER,
                        IN    #OL_I_ID2
INTEGER,
                        IN    #OL_I_ID3
INTEGER,
                        IN    #OL_I_ID4
INTEGER,
                        IN    #OL_I_ID5
INTEGER,
                        IN    #OL_I_ID6
INTEGER,
                        IN    #OL_I_ID7
INTEGER,
                        IN    #OL_I_ID8
INTEGER,
                        IN    #OL_I_ID9
INTEGER,
                        IN    #OL_I_ID10
INTEGER,
                        IN    #OL_I_ID11
INTEGER,
                        IN    #OL_I_ID12
INTEGER,
                        IN    #OL_I_ID13
INTEGER,
                        IN    #OL_I_ID14
INTEGER,
                        IN    #OL_I_ID15
INTEGER,
                        IN    #OL_I_ID_JOIN
VARCHAR(105) ,
                        IN    #OL_QUANTITY_JOIN
VARCHAR(60)  ,
                        INOUT #S_JOIN
VARCHAR(1215),
                        INOUT #I_JOIN
VARCHAR(1215),
                        IN    #OL_SUPPLY_W_JOIN
VARCHAR(60)
                      )
NEWORDER:BEGIN
-- DECLARE
    DECLARE SQLSTATE         CHAR(5)
DEFAULT '00000';
    DECLARE SAPSTOP          CHAR(1)
DEFAULT '/';
    DECLARE @OL_I_ID         INTEGER;
    DECLARE @OL_SUPPLY_W_ID  SMALLINT;
    DECLARE @OL_QUANTITY     SMALLINT;
    DECLARE @S_QUANTITY      SMALLINT;
    DECLARE @I_PRICEH        SMALLINT;
    DECLARE @I_NAMEH         CHAR(24);
    DECLARE @I_DATAH         CHAR(50);
    DECLARE @S_DATA          CHAR(50);
    DECLARE @S_YTD           INTEGER;
    DECLARE @S_ORDER_CNT     SMALLINT;
    DECLARE @S_REMOTE_CNT    SMALLINT;
    DECLARE @D_NEXT_O_ID     INTEGER;
    DECLARE @OL_NUMBER       SMALLINT;
    DECLARE @STOCK_NUM       SMALLINT;
    DECLARE @MATCH_TBL_CNT   SMALLINT;
    DECLARE @S_DIST          CHAR(24);
    DECLARE @S_DIST_01       CHAR(24);
    DECLARE @S_DIST_02       CHAR(24);
    DECLARE @S_DIST_03       CHAR(24);
    DECLARE @S_DIST_04       CHAR(24);
    DECLARE @S_DIST_05       CHAR(24);
    DECLARE @S_DIST_06       CHAR(24);
    DECLARE @S_DIST_07       CHAR(24);
    DECLARE @S_DIST_08       CHAR(24);
    DECLARE @S_DIST_09       CHAR(24);
    DECLARE @S_DIST_10       CHAR(24);
    DECLARE @S_DIST_JOIN     CHAR(240)  ;
    DECLARE @C_OL_I_ID       CHAR(7)    ;
    DECLARE @C_I_PRICEH      CHAR(6)    ;
    DECLARE @C_S_QUANTITY    CHAR(6)    ;
    DECLARE @OL_AMOUNT       INTEGER    ;
    DECLARE @O_OL_CNT        SMALLINT   ;
    DECLARE @DIST_POS        SMALLINT   ;
    DECLARE @FILL_CNT        SMALLINT;
    DECLARE @TMP_CNT         SMALLINT;

-- (7) ITEM     table sele(IN)
    DECLARE ITEM_H CURSOR FOR
        SELECT I_PRICE,
            I_NAME,
            I_DATA,
            I_ID
        FROM   TPCC_SCHEMA.ITEM
        WHERE  TPCC_SCHEMA.ITEM.I_ID
                    IN( #OL_I_ID1    ,
                      #OL_I_ID2    ,
                      #OL_I_ID3    ,
                      #OL_I_ID4    ,
                      #OL_I_ID5    ,
                      #OL_I_ID6    ,
                      #OL_I_ID7    ,
                      #OL_I_ID8    ,
                      #OL_I_ID9    ,
                      #OL_I_ID10   ,
                      #OL_I_ID11   ,
                      #OL_I_ID12   ,
                      #OL_I_ID13   ,
                      #OL_I_ID14   ,
                      #OL_I_ID15   ) ;

-- (8) STOCK    table select
    DECLARE CNSS_HOME CURSOR FOR
        SELECT S_I_ID,S_QUANTITY,
            S_DIST_01,S_DIST_02,S_DIST_03,S_DI
ST_04,S_DIST_05,
            S_DIST_06,S_DIST_07,S_DIST_08,S_DI
ST_09,S_DIST_10,
            S_YTD,S_ORDER_CNT,S_REMOTE_C
NT,S_DATA
          FROM  TPCC_SCHEMA.STOCK
          WHERE S_W_ID = #W_ID
           AND S_I_ID  IN( #OL_I_ID1    ,
                    #OL_I_ID2    ,
                    #OL_I_ID3    ,
                    #OL_I_ID4    ,
                    #OL_I_ID5    ,
                    #OL_I_ID6    ,
                    #OL_I_ID7    ,
                    #OL_I_ID8    ,
```

```
                    #OL_I_ID9     ,
                    #OL_I_ID10    ,
                    #OL_I_ID11    ,
                    #OL_I_ID12    ,
                    #OL_I_ID13    ,
                    #OL_I_ID14    ,
                    #OL_I_ID15    )
        ORDER BY S_I_ID DESC
          FOR UPDATE ;

    SET @DIST_POS    = 1+((#D_ID-1)*24);
    SET @O_OL_CNT    = #H_CNT + #R_CNT ;
    SET #O_ALL_LOCAL = 1               ;

    -- (4) CUSTOMER  table select
                        WHENEVER SQLERROR
GOTO ERR_S_CM;
                        WHENEVER NOT FOUND
GOTO ERR_S_CM;
    SELECT C_LAST,C_CREDIT,C_DISCOUNT
        INTO  #C_LAST,
            #C_CREDIT,
            #C_DISCOUNT
        FROM  TPCC_SCHEMA.CUSTOMER
        WHERE C_W_ID = #W_ID
          AND C_D_ID = #D_ID
          AND C_ID   = #C_ID;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
    IF  #H_CNT = 0 THEN
      GOTO REMORT_PROC ;
    END IF;

HOME_PROC:
-- Home Warehouse PROCESS START
--  (                Warehouse id                )
    -- (7) ITEM      table select
                        WHENEVER SQLERROR
GOTO ERR_S_IT;
                        WHENEVER NOT FOUND
GOTO ERR_S_IT;
    OPEN  ITEM_H ;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
-- LOOP
    SET   @MATCH_TBL_CNT = 0 ;
    INCNT:LOOP
                        WHENEVER SQLERROR
GOTO ERR_S_IT;
                        WHENEVER NOT FOUND
GOTO L1;
        FETCH  ITEM_H
            INTO  @I_PRICEH,
                @I_NAMEH,
                @I_DATAH,
                @OL_I_ID;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;

        SET   @MATCH_TBL_CNT =
@MATCH_TBL_CNT + 1;
        SET   @C_I_PRICEH    = CAST(@I_PRICEH
AS CHAR(6) )         ;
        SET   #I_JOIN        = #I_JOIN    ||
@C_I_PRICEH ||
                    @I_NAMEH   || @I_DATAH    ||
SAPSTOP ;

    END LOOP INCNT;
-- LOOP END

L1:   IF  @MATCH_TBL_CNT < #H_CNT  THEN
        SET    #ITEM_NF_CTR = -2 ;
        GOTO   NORMAL_END ;
    END IF;

    CLOSE  ITEM_H ;

--      (8) STOCK     table select
--      (9) STOCK     table update
                        WHENEVER SQLERROR
GOTO ERR_S_ST;
                        WHENEVER NOT FOUND
GOTO ERR_S_ST;
    OPEN   CNSS_HOME   ;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
-- LOOP
    SET   @STOCK_NUM = 0;
    OLCNT:LOOP
      IF  @STOCK_NUM = #H_CNT THEN
          GOTO L3 ;
      END IF;

        -- (8) STOCK     table select
                        WHENEVER SQLERROR
GOTO ERR_S_ST;
                        WHENEVER NOT FOUND
GOTO L3      ;
        FETCH  CNSS_HOME
            INTO @OL_I_ID,@S_QUANTITY,
                @S_DIST_01,@S_DIST_02,@S_DIST_0
3,@S_DIST_04,@S_DIST_05,
                @S_DIST_06,@S_DIST_07,@S_DIST_0
8,@S_DIST_09,@S_DIST_10,
                @S_YTD,@S_ORDER_CNT,@S_REMO
TE_CNT,@S_DATA;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
        SET  @S_DIST_JOIN = @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                         ;
        SET  @S_DIST =
SUBSTRING(@S_DIST_JOIN FROM @DIST_POS
FOR 24)    ;
        SET @OL_QUANTITY  =
CAST(SUBSTRING(#OL_QUANTITY_JOIN
                    FROM 1+(@STOCK_NUM * 4)
FOR 4)
                    AS SMALLINT )                ;

        SET @S_QUANTITY = (@S_QUANTITY -
@OL_QUANTITY);
        IF  @S_QUANTITY < 10 THEN
            SET @S_QUANTITY = @S_QUANTITY +
91  ;
        END IF;

        SET @S_YTD       = @S_YTD +
@OL_QUANTITY;
        SET @S_ORDER_CNT = @S_ORDER_CNT +
1;

        -- (9) STOCK     table update
                        WHENEVER SQLERROR
GOTO ERR_U_ST;
        UPDATE TPCC_SCHEMA.STOCK
            SET   S_QUANTITY   = @S_QUANTITY,
                S_YTD        = @S_YTD,
                S_ORDER_CNT  = @S_ORDER_CNT,
                S_REMOTE_CNT = @S_REMOTE_CNT
            WHERE CURRENT OF CNSS_HOME ;
                        WHENEVER SQLERROR
CONTINUE;

        SET   @C_S_QUANTITY  =
CAST(@S_QUANTITY AS CHAR(6) )         ;
        SET   #S_JOIN        = #S_JOIN    ||
@C_S_QUANTITY ||
                    @S_DIST    || @S_DATA       ||
SAPSTOP ;

        SET @STOCK_NUM = @STOCK_NUM + 1;

    END LOOP OLCNT;
-- LOOP END

L3:   IF    @STOCK_NUM <> #H_CNT
      AND @STOCK_NUM <> #ITEM_NF_CTR THEN
        GOTO ERR_S_ST_NF;
    END IF;
    CLOSE  CNSS_HOME   ;

--
-- LOOP
    SET   @FILL_CNT = 0 ;
    SET   @TMP_CNT  = #H_CNT - @STOCK_NUM ;
    FILLCNT:LOOP
      IF  @FILL_CNT = @TMP_CNT THEN
          GOTO L4 ;
      END IF;

      SET   #I_JOIN        = #I_JOIN ||
                    '0     '||
                    '1234567890123456789012345678901234' ||
                    '12345678901234567890123456789012345678901234567890' ||
                    SAPSTOP ;
      SET   #S_JOIN        = #S_JOIN ||
                    '123456' ||
                    '1234567890123456789012345678901234' ||
                    '12345678901234567890123456789012345678901234567890' ||
```

```
                    SAPSTOP ;

    SET @FILL_CNT = @FILL_CNT + 1;

  END LOOP FILLCNT;
-- LOOP END

-- Home Warehouse PROCESS END

L4:  IF  #R_CNT = 0 THEN
     GOTO DISTRICT_PROC ;
   END IF;

REMORT_PROC:
-- Remote Warehouse process start
--  (                        Warehouse)
-- LOOP
   SET   @MATCH_TBL_CNT = 0 ;
   SET   @STOCK_NUM     = 0 ;

   OLCNT_R:LOOP
R1:   IF  @STOCK_NUM = #R_CNT THEN
      GOTO R3 ;
    END IF;

    SET  @OL_I_ID       =
CAST(SUBSTRING(#OL_I_ID_JOIN
                    FROM 1+(@STOCK_NUM * 7)
FOR 7)
                    AS INTEGER )                ;

    SET  @OL_SUPPLY_W_ID =
CAST(SUBSTRING(#OL_SUPPLY_W_JOIN
                    FROM 1+(@STOCK_NUM * 4)
FOR 4)
                    AS SMALLINT )               ;
    -- (7) ITEM     table select
                    WHENEVER SQLERROR
GOTO ERR_S_IT ;
                    WHENEVER NOT FOUND
GOTO R4     ;
    SELECT I_PRICE,I_NAME,I_DATA
        INTO   @I_PRICEH,
               @I_NAMEH ,
               @I_DATAH
        FROM   TPCC_SCHEMA.ITEM
        WHERE  I_ID = @OL_I_ID ;
                    WHENEVER SQLERROR
CONTINUE;
                    WHENEVER NOT FOUND
CONTINUE;

    SET   @MATCH_TBL_CNT =
@MATCH_TBL_CNT + 1    ;

    SET   @C_I_PRICEH    = CAST(@I_PRICEH AS
CHAR(6) )       ;
    SET   #I_JOIN       = #I_JOIN    || @C_I_PRICEH
||
                @I_NAMEH   || @I_DATAH    ||
SAPSTOP ;

    -- (8) STOCK     table select
                    WHENEVER SQLERROR
GOTO ERR_S_ST;
                    WHENEVER NOT FOUND
GOTO ERR_S_ST;
    SELECT  S_QUANTITY,
            S_DIST_01,S_DIST_02,S_DIST_03,S_DIST
_04,S_DIST_05,
            S_DIST_06,S_DIST_07,S_DIST_08,S_DIST
_09,S_DIST_10,
            S_YTD,S_ORDER_CNT,S_REMOTE_CNT,
S_DATA
        INTO @S_QUANTITY,
           @S_DIST_01,@S_DIST_02,@S_DIST_03,@
S_DIST_04,@S_DIST_05,
           @S_DIST_06,@S_DIST_07,@S_DIST_08,@
S_DIST_09,@S_DIST_10,
           @S_YTD,@S_ORDER_CNT,@S_REMOTE_
CNT,@S_DATA
        FROM  TPCC_SCHEMA.STOCK
        WHERE S_W_ID = @OL_SUPPLY_W_ID
          AND S_I_ID = @OL_I_ID  ;
                    WHENEVER SQLERROR
CONTINUE;
                    WHENEVER NOT FOUND
CONTINUE;

        SET  @S_DIST_JOIN = @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                      ;
        SET  @S_DIST =
SUBSTRING(@S_DIST_JOIN FROM @DIST_POS
FOR 24)   ;

        SET  @OL_QUANTITY  =
CAST(SUBSTRING(#OL_QUANTITY_JOIN
                    FROM
1+((@STOCK_NUM+#H_CNT) * 4) FOR 4)
                    AS SMALLINT )               ;

        SET @S_QUANTITY = (@S_QUANTITY -
@OL_QUANTITY);
        IF  @S_QUANTITY < 10 THEN
          SET @S_QUANTITY = @S_QUANTITY +
91  ;
        END IF;

        SET @S_YTD       = @S_YTD +
@OL_QUANTITY;
        SET @S_ORDER_CNT = @S_ORDER_CNT +
1;
        IF  @OL_SUPPLY_W_ID <> #W_ID THEN
          SET @S_REMOTE_CNT =
@S_REMOTE_CNT + 1;
          SET #O_ALL_LOCAL  = 0;
        END IF;

    -- (9) STOCK     table update
                    WHENEVER SQLERROR
GOTO ERR_U_ST;
    UPDATE TPCC_SCHEMA.STOCK
      SET   S_QUANTITY   = @S_QUANTITY,
          S_YTD        = @S_YTD,
          S_ORDER_CNT  = @S_ORDER_CNT,
          S_REMOTE_CNT = @S_REMOTE_CNT
     WHERE   S_W_ID = @OL_SUPPLY_W_ID
       AND   S_I_ID = @OL_I_ID  ;
                    WHENEVER SQLERROR
CONTINUE;

    SET   @C_S_QUANTITY  =
CAST(@S_QUANTITY AS CHAR(6) )       ;
    SET   #S_JOIN       = #S_JOIN    ||
@C_S_QUANTITY ||
                @S_DIST    || @S_DATA      ||
SAPSTOP ;

R2:   SET @STOCK_NUM = @STOCK_NUM + 1 ;

  END LOOP OLCNT_R;

-- LOOP END
R3:
  IF  @MATCH_TBL_CNT < #R_CNT  THEN
    IF #ITEM_NF_CTR = -1 THEN
      SET   #ITEM_NF_CTR =
@MATCH_TBL_CNT ;
    ELSE
      SET   #ITEM_NF_CTR = #ITEM_NF_CTR +
@MATCH_TBL_CNT ;
    END IF;
  END IF;

-- Remote Warehouse process end

DISTRICT_PROC:
  -- (3) DISTRICT  table update
                    WHENEVER SQLERROR
GOTO ERR_U_DI;
                    WHENEVER NOT FOUND
GOTO ERR_U_DI;
  UPDATE TPCC_SCHEMA.DISTRICT
      SET   D_NEXT_O_ID = D_NEXT_O_ID+1
      WHERE D_W_ID      = #W_ID
        AND D_ID        = #D_ID ;
                    WHENEVER SQLERROR
CONTINUE;
                    WHENEVER NOT FOUND
CONTINUE;
  -- (2) DISTRICT  table select

  SELECT  D_NEXT_O_ID-1,D_TAX
      INTO  #O_ID ,#D_TAX
      FROM  TPCC_SCHEMA.DISTRICT
      WHERE D_W_ID = #W_ID
        AND D_ID   = #D_ID ;

  -- (6) ORDERS    table insert
                    WHENEVER SQLERROR
GOTO ERR_I_OR;
                    WHENEVER NOT FOUND
GOTO ERR_I_OR;
  INSERT INTO TPCC_SCHEMA.ORDERS
      VALUES (#O_ID,
              #D_ID,
              #W_ID,
              #C_ID,
              #O_ENTRY_D,
              NULL,
              @O_OL_CNT,
              #O_ALL_LOCAL);
                    WHENEVER SQLERROR
CONTINUE;
```

```
                        WHENEVER NOT FOUND
CONTINUE;
    -- (5) NEWORDER  table insert
                        WHENEVER SQLERROR
GOTO ERR_I_NO;
                        WHENEVER NOT FOUND
GOTO ERR_I_NO;
    INSERT INTO TPCC_SCHEMA.NEWORDER
          VALUES (#O_ID,
                #D_ID,
                #W_ID);
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
    -- (1) WAREHOUSE table update
                        WHENEVER SQLERROR
GOTO ERR_S_WH;
    SELECT W_TAX
          INTO  #W_TAX
          FROM  TPCC_SCHEMA.WAREHOUSE
          WHERE W_ID=#W_ID ;
                        WHENEVER SQLERROR
CONTINUE;

NORMAL_END:
    SET #STATE  = '00000' ;
    LEAVE NEWORDER ;

--
--
R4:    SET   #I_JOIN       = #I_JOIN ||
                        '0    '||
                        '12345678901234567890123' ||
                        '12345678901234567890123456
901234567890123456789' ||
                        SAPSTOP ;
      SET   #S_JOIN       = #S_JOIN ||
                        '123456' ||
                        '12345678901234567890123' ||
                        '12345678901234567890123456
901234567890123456789' ||
                        SAPSTOP ;
    GOTO R2 ;


--SQLERR:NOT_FOUND:
ERR_I_OR:
      SET #ERRPOS = 107    ;
      SET #STATE  = SQLSTATE;
      LEAVE NEWORDER       ;
ERR_I_OL:
      SET #ERRPOS = 108    ;
      SET #STATE  = SQLSTATE;
      LEAVE NEWORDER       ;
ERR_I_NO:
      SET #ERRPOS = 109    ;
      SET #STATE  = SQLSTATE;
      LEAVE NEWORDER       ;
ERR_S_IT:
      SET #ERRPOS = 201    ;
      SET #STATE  = SQLSTATE;
      LEAVE NEWORDER       ;
ERR_S_WH:
      SET #ERRPOS = 202    ;
      SET #STATE  = SQLSTATE;
      LEAVE NEWORDER       ;
ERR_S_DI:
      SET #ERRPOS = 203    ;
      SET #STATE  = SQLSTATE;
      LEAVE NEWORDER       ;
ERR_S_ST:
      SET #ERRPOS = 204    ;
      SET #STATE  = SQLSTATE;
      LEAVE NEWORDER       ;
ERR_S_ST_NF:
      SET #ERRPOS = 204    ;
      SET #STATE  = '02000' ;
      LEAVE NEWORDER       ;
ERR_S_CM:
      SET #ERRPOS = 205    ;
      SET #STATE  = SQLSTATE;
      LEAVE NEWORDER       ;
ERR_U_DI:
      SET #ERRPOS = 303    ;
      SET #STATE  = SQLSTATE;
      LEAVE NEWORDER       ;
ERR_U_ST:
      SET #ERRPOS = 304    ;
      SET #STATE  = SQLSTATE;

END NEWORDER
END-EXEC;
```

File: Y_ODERSTAT

```
-- /**********STORED
PROCEDURE***********************************************/
-- /** Y_ODERSTAT   COPYRIGHT FUJITSU LIMITED
1997                **/
-- /**    :                                          **/
-- /**    :                                          **/
-- /**    :    SymfoWARE RDB TPC-C Benchmark
**/
-- /**    :   Order-Status                           **/
-- /**    :    1996/10/12                            **/
-- /**         1997/03/13  Revision3.3 : Any Error(Clause
2.3.6)    **/
--
/*****************************************************************
********/

--  #OL_JOIN            VARCHAR(570)
--  +-----------------------------------+
--  | sqllen       short                |
--  +-sqlver-----------------------------+  ---
--  | #OL_I_IDn      CHAR(7)          |  |
--  +-----------------------------------+  |
--  | #OL_AMOUNTn    CHAR(8)          |  |
--  +-----------------------------------+  |
--  | #OL_SUPPLY_W_IDn CHAR(4)          |  |
--  +-----------------------------------+  |
--  | #OL_QUANTITYn  CHAR(4)          |  |
--  +-----------------------------------+  |
--  | #OL_DELIVERYn  CHAR(14)         |  |
--  +-----------------------------------+  |
--  |              CHAR(1) "/"       |  |
--  +-----------------------------------+  ---
--
--  +-----------------------------------+
--
--
--
*****************************************************************
*************
```

```
EXEC SQL
CREATE PROCEDURE
TPCC_SCHEMA.Y_ODERSTAT(OUT #STATE
CHAR(5) ,
                        INOUT #ERRPOS
INTEGER ,
                        IN    #W_ID
SMALLINT,
                        IN    #D_ID          SMALLINT,
                        INOUT #C_ID
INTEGER ,
                        OUT   #C_FIRST
CHAR(16),
                        OUT   #C_MIDDLE
CHAR(2) ,
                        INOUT #C_LAST
CHAR(16),
                        OUT   #C_BALANCE
DOUBLE PRECISION,
                        INOUT #O_ID
INTEGER ,
                        OUT   #O_ENTRY_D
CHAR(14),
                        OUT   #O_CARRIER_ID
SMALLINT,
                        INOUT #O_OL_CNT
SMALLINT,
                        INOUT #OL_JOIN
VARCHAR(570)
                              )
ORDER_STATUS:BEGIN
-- DECLARE
    DECLARE SQLSTATE          CHAR(5)
DEFAULT '00000';
    DECLARE SAPSTOP           CHAR(1)
DEFAULT '/';
    DECLARE DELIVERY_D        CHAR(14)
DEFAULT '777777777';
    DECLARE @OL_I_ID          INTEGER;
    DECLARE @OL_SUPPLY_W_ID   SMALLINT;
    DECLARE @OL_QUANTITY      SMALLINT;
    DECLARE @OL_AMOUNT        INTEGER;
    DECLARE @OL_DELIVERY_D    CHAR(14);
    DECLARE @OL_NUMBER        INTEGER;
    DECLARE @NAMECOUNT        INTEGER;
    DECLARE @J                INTEGER;
    DECLARE @I                INTEGER;
    DECLARE @WORK             VARCHAR(100);

-- DEFINE  CUSTOMER  table cursor
   DECLARE COCS  CURSOR FOR
         SELECT C_ID,
               C_FIRST,
               C_MIDDLE,
               C_LAST,
               C_BALANCE
               FROM  TPCC_SCHEMA.CUSTOMER
               WHERE C_LAST = #C_LAST
                 AND C_W_ID = #W_ID
                 AND C_D_ID = #D_ID
               ORDER BY C_FIRST;

-- DEFINE  ORDERLINE table cursor
    DECLARE COOLS CURSOR FOR
          SELECT OL_I_ID,
               OL_SUPPLY_W_ID,
               OL_DELIVERY_D,
               OL_QUANTITY,
```

```
            OL_AMOUNT
            FROM  TPCC_SCHEMA.ORDERLINE
            WHERE OL_W_ID = #W_ID
              AND OL_D_ID = #D_ID
              AND OL_O_ID = #O_ID
              AND OL_NUMBER
IN(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15);
--$
--$              WHERE OL_W_ID = #W_ID
--$                AND OL_D_ID = #D_ID
--$                AND OL_O_ID = #O_ID;

    IF #C_ID = 0 THEN
      -- Customer Last Name Payment Transaction
      -- (1) CUSTOMER  table select
                        WHENEVER SQLERROR
GOTO ERR_S_CM;
                        WHENEVER NOT FOUND
GOTO ERR_S_CM;
       SELECT COUNT(*)
            INTO  @NAMECOUNT
         FROM  TPCC_SCHEMA.CUSTOMER
         WHERE C_LAST = #C_LAST
          AND C_W_ID = #W_ID
          AND C_D_ID = #D_ID;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
       IF @NAMECOUNT > 0 THEN
                        WHENEVER SQLERROR
GOTO ERR_S_CM;
                        WHENEVER NOT FOUND
GOTO ERR_S_CM;
         OPEN  COCS;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
         SET   @J = @NAMECOUNT + 1;
         SET   @J = @J       / 2;
         SET   @I = 0          ;
         NAMECNT:LOOP
           IF  @I = @J THEN
             LEAVE NAMECNT ;
           END IF;
           SET   @I = @I + 1  ;
                        WHENEVER SQLERROR
GOTO ERR_S_CM;
                        WHENEVER NOT FOUND
GOTO ERR_S_CM;
           FETCH  COCS
                INTO #C_ID,
                   #C_FIRST,
                   #C_MIDDLE,
                   #C_LAST,
                   #C_BALANCE;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
         END LOOP NAMECNT;
         CLOSE COCS;
       ELSE
         GOTO ERR_S_CM_NAME ;
       END IF;

    ELSE
      -- Customer id Payment Transaction
      -- (2) CUSTOMER  table select
                        WHENEVER SQLERROR
GOTO ERR_S_CM;
                        WHENEVER NOT FOUND
GOTO ERR_S_CM;
       SELECT
C_FIRST,C_MIDDLE,C_LAST,C_BALANCE
            INTO  #C_FIRST,
                #C_MIDDLE,
                #C_LAST,
                #C_BALANCE
            FROM  TPCC_SCHEMA.CUSTOMER
            WHERE C_ID   = #C_ID
              AND C_D_ID = #D_ID
              AND C_W_ID = #W_ID;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
    END IF;

-- (3) ORDER     table select get max o_id record
                        WHENEVER SQLERROR
GOTO ERR_S_OR;
                        WHENEVER NOT FOUND
GOTO ERR_S_OR;
    SELECT   O_ID,
           O_ENTRY_D,
           O_CARRIER_ID,
           O_OL_CNT
           INTO  #O_ID,
               #O_ENTRY_D,
               #O_CARRIER_ID,
               #O_OL_CNT
           FROM  TPCC_SCHEMA.ORDERS
           WHERE O_ID   = (SELECT MAX(O_ID)
                           FROM
TPCC_SCHEMA.ORDERS
                           WHERE O_W_ID = #W_ID
                             AND O_D_ID = #D_ID
                             AND O_C_ID = #C_ID )
             AND O_W_ID = #W_ID
             AND O_D_ID = #D_ID
             AND O_C_ID = #C_ID;
                        WHENEVER SQLERROR
GOTO ERR_S_OL;
                        WHENEVER NOT FOUND
GOTO ERR_S_OL;
    OPEN     COOLS ;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
-- LOOP
    SET   @OL_NUMBER = 1;
    OLCNT:LOOP
      IF  #O_OL_CNT < @OL_NUMBER THEN
        LEAVE OLCNT ;
      END IF;

    -- (4) ORDER-LINE table select
                        WHENEVER SQLERROR
GOTO ERR_S_OL;
                        WHENEVER NOT FOUND
GOTO ERR_S_OL;
         FETCH  COOLS
              INTO  @OL_I_ID,
                  @OL_SUPPLY_W_ID,
                  @OL_DELIVERY_D,
                  @OL_QUANTITY,
                  @OL_AMOUNT;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
       IF  @OL_DELIVERY_D IS NULL THEN
         SET @WORK    = CAST(@OL_I_ID       AS
CHAR(7))
                || CAST(@OL_AMOUNT     AS
CHAR(8))
                || CAST(@OL_SUPPLY_W_ID AS
CHAR(4))
                || CAST(@OL_QUANTITY    AS
CHAR(4))
                ||         DELIVERY_D  ||
SAPSTOP   ;
       ELSE
         SET @WORK    = CAST(@OL_I_ID       AS
CHAR(7))
                || CAST(@OL_AMOUNT     AS
CHAR(8))
                || CAST(@OL_SUPPLY_W_ID AS
CHAR(4))
                || CAST(@OL_QUANTITY    AS
CHAR(4))
                ||      @OL_DELIVERY_D  ||
SAPSTOP   ;
       END IF ;

       SET #OL_JOIN = #OL_JOIN || @WORK;

       SET @OL_NUMBER  = @OL_NUMBER + 1;

    END LOOP OLCNT;
-- LOOP END

    CLOSE COOLS         ;

    COMMIT WORK         ;
    SET #STATE = '00000' ;
    LEAVE ORDER_STATUS   ;

--SQLERR:NOT_FOUND:
ERR_S_CM_NAME:
    SET #ERRPOS = 205    ;
    SET #STATE  = '02000' ;
    ROLLBACK WORK        ;
    LEAVE ORDER_STATUS   ;
ERR_S_CM:
    SET #ERRPOS = 205    ;
    SET #STATE  = SQLSTATE;
    ROLLBACK WORK        ;
    LEAVE ORDER_STATUS   ;
ERR_S_OR:
    SET #ERRPOS = 207    ;
    SET #STATE  = SQLSTATE;
    ROLLBACK WORK        ;
    LEAVE ORDER_STATUS   ;
ERR_S_OL:
    SET #ERRPOS = 208    ;
    SET #STATE  = SQLSTATE;
    ROLLBACK WORK        ;

END ORDER_STATUS
END-EXEC;
```

File: Y_PAYMENT_cent

```
-- /**********STORED
PROCEDURE*********************************************/
-- /** Y_PAYMENT    COPYRIGHT FUJITSU LIMITED
1997                **/
-- /**    :                                          **/
-- /**    :                                          **/
-- /**    :    SymfoWARE RDB TPC-C Benchmark
**/
-- /**    :   Payment                                **/
-- /**    :    1996/10/12                            **/
-- /**        1997/03/13  Revision3.3 : Any Error(Clause
2.3.6)     **/
--
/*****************************************************************
********/

EXEC SQL
CREATE PROCEDURE
TPCC_SCHEMA.Y_PAYMENT(OUT #STATE
CHAR(5) ,
                    INOUT #ERRPOS
INTEGER ,
                    IN   #W_ID          SMALLINT,
                    IN   #D_ID          SMALLINT,
                    INOUT #C_ID
INTEGER ,
                    IN   #C_D_ID
SMALLINT,
                    IN   #C_W_ID
SMALLINT,
                    IN   #H_AMOUNT
INTEGER ,
                    IN   #H_DATE
CHAR(14),
                    INOUT #W_NAME
CHAR(10),
                    OUT  #W_STREET_1
CHAR(20),
                    OUT  #W_STREET_2
CHAR(20),
                    OUT  #W_CITY
CHAR(20),
                    OUT  #W_STATE
CHAR(2),
                    OUT  #W_ZIP
CHAR(9),
                    INOUT #D_NAME
CHAR(10),
                    OUT  #D_STREET_1
CHAR(20),
                    OUT  #D_STREET_2
CHAR(20),
                    OUT  #D_CITY
CHAR(20),
                    OUT  #D_STATE
CHAR(2),
                    OUT  #D_ZIP         CHAR(9),
                    OUT  #C_FIRST
CHAR(16),
                    OUT  #C_MIDDLE
CHAR(2),
                    INOUT #C_LAST
CHAR(16),
                    OUT  #C_STREET_1
CHAR(20),
                    OUT  #C_STREET_2
CHAR(20),
                    OUT  #C_CITY
CHAR(20),
                    OUT  #C_STATE
CHAR(2),
                    OUT  #C_ZIP         CHAR(9),
                    OUT  #C_PHONE
CHAR(16),
                    INOUT #C_CREDIT
CHAR(2),
                    OUT  #C_CREDIT_LIM
DECIMAL(12,0), --98.11.06
                    OUT  #C_DISCOUNT
SMALLINT,
                    INOUT #C_BALANCE
DECIMAL(12,0), --98.11.06
                    INOUT #C_YTD_PAYMENT
DECIMAL(12,0), --98.11.06
                    INOUT #C_PAYMENT_CNT
SMALLINT,
                    OUT  #C_SINCE
CHAR(14),
                    INOUT #C_DATA
VARCHAR(500)
                   )

PAYMENT:BEGIN
-- DECLARE
     DECLARE SQLSTATE       CHAR(5)
DEFAULT '00000';
     DECLARE @CNT           INTEGER;
     DECLARE @NAMECOUNT     INTEGER;
     DECLARE @W_YTD         DECIMAL(12,0);  --
98.11.06
     DECLARE @D_YTD         DECIMAL(12,0);  --
98.11.06
     DECLARE @C_DATA474     CHAR(474);     --
98.10.13 change
     DECLARE @H_DATA        CHAR(24);
     DECLARE @H_AMOUNT      DECIMAL(10,0);
--98.11.06      +oza

-- CUSTOMER
     DECLARE CPCS CURSOR FOR
          SELECT C_ID,
               C_FIRST,
               C_MIDDLE,
               C_LAST,
               C_STREET_1,
               C_STREET_2,
               C_CITY,
               C_STATE,
               C_ZIP,
               C_PHONE,
               C_SINCE,
               C_CREDIT,
               C_CREDIT_LIM,
               C_DISCOUNT,
               C_BALANCE,
               C_YTD_PAYMENT,
               C_PAYMENT_CNT
          FROM  TPCC_SCHEMA.CUSTOMER
          WHERE C_LAST = #C_LAST
            AND C_W_ID = #C_W_ID
            AND C_D_ID = #C_D_ID
          ORDER BY C_FIRST;

     IF #C_ID = 0 THEN
        -- Customer Last Name process
        -- (5) CUSTOMER   table select
                    WHENEVER SQLERROR
GOTO ERR_S_CM;
                    WHENEVER NOT FOUND
GOTO ERR_S_CM;
        SELECT COUNT(*) INTO @NAMECOUNT
            FROM TPCC_SCHEMA.CUSTOMER
            WHERE C_LAST = #C_LAST
            AND C_W_ID = #C_W_ID
            AND C_D_ID = #C_D_ID;
                    WHENEVER SQLERROR
CONTINUE;
                    WHENEVER NOT FOUND
CONTINUE;
        -- (6) CUSTOMER
        -- Customer Last Name                C_FIRST
        -- NAMECOUNT/
        IF @NAMECOUNT > 0 THEN
          SET  @CNT      = @NAMECOUNT + 1;
          SET  @CNT      = @CNT      / 2;
          SET  @NAMECOUNT = @CNT        ;
                    WHENEVER SQLERROR
GOTO ERR_S_CM;
                    WHENEVER NOT FOUND
GOTO ERR_S_CM;
          OPEN  CPCS;
                    WHENEVER SQLERROR
CONTINUE;
                    WHENEVER NOT FOUND
CONTINUE;
          SET   @CNT   = 0 ;
          WHILE  @CNT < @NAMECOUNT DO
            SET  @CNT = @CNT + 1;
            -- (6) CUSTOMER  table
                    WHENEVER SQLERROR
GOTO ERR_S_CM;
                    WHENEVER NOT FOUND
GOTO ERR_S_CM;
            FETCH CPCS
                INTO #C_ID,
                    #C_FIRST,
                    #C_MIDDLE,
                    #C_LAST,
                    #C_STREET_1,
                    #C_STREET_2,
                    #C_CITY,
                    #C_STATE,
                    #C_ZIP,
                    #C_PHONE,
                    #C_SINCE,
                    #C_CREDIT,
                    #C_CREDIT_LIM,
                    #C_DISCOUNT,
                    #C_BALANCE,
                    #C_YTD_PAYMENT,
                    #C_PAYMENT_CNT;
                    WHENEVER SQLERROR
CONTINUE;
                    WHENEVER NOT FOUND
CONTINUE;
          END WHILE;

          CLOSE CPCS;
```

```
        ELSE
            GOTO ERR_S_CM_NAME;
        END IF;
    ELSE
        -- C-ID PROCESS
        -- (7) CUSTOMER  teble
                        WHENEVER SQLERROR
GOTO ERR_S_CM;
                        WHENEVER NOT FOUND
GOTO ERR_S_CM;
        SELECT C_FIRST,
            C_MIDDLE,
            C_LAST,
            C_STREET_1,
            C_STREET_2,
            C_CITY,
            C_STATE,
            C_ZIP,
            C_PHONE,
            C_SINCE,
            C_CREDIT,
            C_CREDIT_LIM,
            C_DISCOUNT,
            C_BALANCE,
            C_YTD_PAYMENT,
            C_PAYMENT_CNT
         INTO  #C_FIRST,
            #C_MIDDLE,
            #C_LAST,
            #C_STREET_1,
            #C_STREET_2,
            #C_CITY,
            #C_STATE,
            #C_ZIP,
            #C_PHONE,
            #C_SINCE,
            #C_CREDIT,
            #C_CREDIT_LIM,
            #C_DISCOUNT,
            #C_BALANCE,
            #C_YTD_PAYMENT,
            #C_PAYMENT_CNT
         FROM  TPCC_SCHEMA.CUSTOMER
         WHERE C_W_ID = #C_W_ID
          AND C_D_ID = #C_D_ID
          AND C_ID   = #C_ID;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
    END IF;



--
    SET @H_AMOUNT = #H_AMOUNT ;

    -- Customer           #C_BALANCE
    -- Customer               #C_YTD_PAYMENT
    -- Customer               #C_PAYMENT_CNT
    SET #C_BALANCE    = #C_BALANCE    -
@H_AMOUNT ;
    SET #C_YTD_PAYMENT = #C_YTD_PAYMENT
+ @H_AMOUNT ;
    SET #C_PAYMENT_CNT = #C_PAYMENT_CNT
+ 1       ;

    --(8) HISTORY teble insert
    IF #C_CREDIT = 'BC' THEN
        --***************--
        -- Bad Customer
        --***************--
        -- (8)BC-1 CUSTOMER table select
                        WHENEVER SQLERROR
GOTO ERR_S_CM;
                        WHENEVER NOT FOUND
GOTO ERR_S_CM;
        SELECT C_DATA
            INTO  @C_DATA474
            FROM  TPCC_SCHEMA.CUSTOMER
            WHERE C_ID   = #C_ID
             AND C_D_ID = #C_D_ID
             AND C_W_ID = #C_W_ID;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
        -- (8)BC-2      c_data            .
        SET   #C_DATA   = CAST(#C_ID    AS
CHAR(5))
                    || CAST(#C_D_ID   AS CHAR(2))
                    || CAST(#C_W_ID   AS CHAR(4))  -
-98.10.13 change
                    || CAST(#D_ID     AS CHAR(2))
                    || CAST(#W_ID     AS CHAR(4))  --
98.10.13 change
                    || CAST(#H_AMOUNT AS CHAR(7))
                    || ' '
                    || @C_DATA474;


        -- (8) BC-3 CUSTOMER table update
                        WHENEVER SQLERROR
GOTO ERR_U_CM;
                        WHENEVER NOT FOUND
GOTO ERR_U_CM;
        UPDATE   TPCC_SCHEMA.CUSTOMER
           SET  C_BALANCE    = #C_BALANCE,
                C_YTD_PAYMENT =
#C_YTD_PAYMENT,
                C_PAYMENT_CNT =
#C_PAYMENT_CNT,
                C_DATA       = #C_DATA
           WHERE C_ID        = #C_ID
            AND C_D_ID       = #C_D_ID
            AND C_W_ID       = #C_W_ID;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
    ELSE
        --***************--
        -- Good Customer
        --***************--
        -- (8)GC-1 CUSTOMER table update
                        WHENEVER SQLERROR
GOTO ERR_U_CM;
                        WHENEVER NOT FOUND
GOTO ERR_U_CM;
        UPDATE TPCC_SCHEMA.CUSTOMER
            SET  C_BALANCE    = #C_BALANCE,
                 C_YTD_PAYMENT =
#C_YTD_PAYMENT,
                 C_PAYMENT_CNT =
#C_PAYMENT_CNT
            WHERE C_ID   = #C_ID
             AND C_D_ID = #C_D_ID
             AND C_W_ID = #C_W_ID;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
    END IF;

    -- (3) DISTRICT table select
                        WHENEVER SQLERROR
GOTO ERR_S_DI;
                        WHENEVER NOT FOUND
GOTO ERR_S_DI;
    SELECT   D_NAME,
            D_STREET_1,
            D_STREET_2,
            D_CITY,
            D_STATE,
            D_ZIP,
            D_YTD
       INTO  #D_NAME,
            #D_STREET_1,
            #D_STREET_2,
            #D_CITY,
            #D_STATE,
            #D_ZIP,
            @D_YTD
       FROM  TPCC_SCHEMA.DISTRICT
       WHERE D_ID   = #D_ID
        AND D_W_ID = #W_ID;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
    -- (4) DISTRICT
    SET    @D_YTD = @D_YTD + @H_AMOUNT;
                        WHENEVER SQLERROR
GOTO ERR_U_DI;
                        WHENEVER NOT FOUND
GOTO ERR_U_DI;
    UPDATE  TPCC_SCHEMA.DISTRICT
           SET  D_YTD  = @D_YTD
           WHERE D_ID   = #D_ID
            AND D_W_ID = #W_ID;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
    -- (1) WAREHOUSE
                        WHENEVER SQLERROR
GOTO ERR_S_WH;
    SELECT  W_NAME,
            W_STREET_1,
            W_STREET_2,
            W_CITY,
            W_STATE,
            W_ZIP,
            W_YTD
       INTO #W_NAME,
            #W_STREET_1,
            #W_STREET_2,
            #W_CITY,
            #W_STATE,
            #W_ZIP,
            @W_YTD
       FROM  TPCC_SCHEMA.WAREHOUSE
       WHERE W_ID = #W_ID;
                        WHENEVER SQLERROR
CONTINUE;
    -- (2) WAREHOUSE
```

```
      SET    @W_YTD = @W_YTD + @H_AMOUNT;
                          WHENEVER SQLERROR
GOTO ERR_U_WH;
      UPDATE  TPCC_SCHEMA.WAREHOUSE
           SET   W_YTD = @W_YTD
           WHERE W_ID  = #W_ID;
                          WHENEVER SQLERROR
CONTINUE;
      -- (9) HISTORY                        #H_DATA
      --    HISTORY
      SET    @H_DATA = #W_NAME || '    ' ||
#D_NAME;
                          WHENEVER SQLERROR
GOTO ERR_I_HI;
                          WHENEVER NOT FOUND
GOTO ERR_I_HI;
      INSERT
           INTO   TPCC_SCHEMA.HISTORY
                (H_C_ID,
                 H_C_D_ID,
                 H_C_W_ID,
                 H_D_ID,
                 H_W_ID,
                 H_DATE,
                 H_AMOUNT,
                 H_DATA)
           VALUES (#C_ID,
                 #C_D_ID,
                 #C_W_ID,
                 #D_ID,
                 #W_ID,
                 #H_DATE,
                 #H_AMOUNT,
                 @H_DATA);
                          WHENEVER SQLERROR
CONTINUE;
                          WHENEVER NOT FOUND
CONTINUE;
    COMMIT WORK         ;
    SET #STATE = '00000' ;
    LEAVE PAYMENT       ;

--SQLERR:NOT_FOUND:
ERR_I_HI:
      SET #ERRPOS = 106    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;
      LEAVE PAYMENT        ;
ERR_S_WH:
      SET #ERRPOS = 202    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;
      LEAVE PAYMENT        ;
ERR_S_DI:
      SET #ERRPOS = 203    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;
      LEAVE PAYMENT        ;
ERR_S_CM_NAME:
      SET #ERRPOS = 205    ;
      SET #STATE  = '02000' ;
      ROLLBACK WORK        ;
      LEAVE PAYMENT        ;
ERR_S_CM:
      SET #ERRPOS = 205    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;
      LEAVE PAYMENT        ;
ERR_U_WH:
      SET #ERRPOS = 302    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;
      LEAVE PAYMENT        ;
ERR_U_DI:
      SET #ERRPOS = 303    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;
      LEAVE PAYMENT        ;
ERR_U_CM:
      SET #ERRPOS = 305    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;

END PAYMENT
END-EXEC;
```

Appendix C: RTE Scripts

File: tpcc

```
#!/usr/bin/sh

sh rmlog

date
date > connect_time

#nawk -f val.awk ../data/t > ../data/tpcC.t.edt
echo "exec psinit"

./psinit -f ../data/t -t 1 -X  # up-date point

echo "psinit end"
date >> connect_time
date
```

File: tpcc.conf

```
#
#  tpcC.conf : configration file for TPC-C
#
#

STARTGROUP = sync , 1
      STARTRTE
            RTEHOST    = rte04
            STARTSUT
                  SUTHOST    = cl14a,420
                  SUTLOGIN  = oracle
                  SUTPASSWD = oracle
                  SUTCMD    = Tc
            ENDSUT
      ENDRTE
#      STRCMD    = tpcCstartCmdSH
#      TSCOM     = tpcCtscomSH
#      TECOM     = tpcCtecomSH
      LOGOUT    = NONE
      LOGMODE   = ALL
      LOGCOMMENT= COMOFF
      LOGFILE   = tpcC.log
      SIMFILE   = ../data/tpcc.pps
      PROTCOL = telnet,80
#WAREHOUSE SCALE
      VAL       = U1I = 2640
#RAMP-UP TIME
      VAL       = U2I = 0
#MEASUERMENT TIME
      VAL       = U3I = 3000
#RAMP-DOWN TIME
      VAL       = U4I = 0
#NEW THINKTIME (msec)
      VAL       = U5I = 12040
#PAY THINKTIME (msec)
      VAL       = U6I = 12040
#
      VAL       = U7I = 0
      VAL       = U8I = 0
      VAL       = U9I = 0
#
#ORD THINKTIME (msec)
      VAL       = U10I = 10190
#DEL THINKTIME (msec)
      VAL       = U11I = 5040
#STK THINKTIME (msec)
      VAL       = U12I = 5040
#NURAND CONSTANT c_id
      VAL       = U13I = 777
#NURAND CONSTANT c_last
      VAL       = U14I = 111
#NURAND CONSTANT ol_i_id
      VAL       = U15I = 3562
#MSG  OFF:0, Each Term:1, Field:2
      VAL       = U16I= 0
#NEW KEYING-TIME (msec)
      VAL       = U17I = 18100
#PAY KEYING-TIME (msec)
      VAL       = U18I = 3050
#ORD KEYING-TIME (msec)
      VAL       = U19I = 2050
#DEL KEYING-TIME (msec)
      VAL       = U20I = 2050
#STK KEYING-TIME (msec)
      VAL       = U21I = 2050
ENDGROUP
```

Appendix D: System Tunables

File: Client.reg

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu]

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\ADJUST]

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\ADJUST\Adjust]
"insdir"="C:\\ADJUST\\"
"ver"="V2.1L30"
"JefConvTbl"=dword:00000000

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\ADJUST\CONVERT TYPE]
"EUCTYPE"="U90"
"JEF"="JEFAUG"
"JIS"="JISKANA"
"S-JISTYPE"="R90"
"UDEFCTL"="USE"

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\ADJUST\OFFSET]
"JEFOFFSET"=""
"S-EUCOFFSET"=""
"S-JISOFFSET"=""

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\ADJUST\V2.1]

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\Install]

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\Install\ADJUST]
"Directory"="C:\\ADJUST\\"
"VersionLevel"="V2.1L30"
"LogFileLink"="C:\\ADJUST\\UnInAdj.log"
"Install_Date"=hex:94,1b,01,85,7f,15,59,17,19,1c,
04,1f,0d,69,29,b5,df,99,47,83,\
  31,8a,11,29,7d,99,17,0c,1e,53,1a,25,1e,67,d4,c0,
f4,7d,c2,25,14,80,1a,0f,5b,\
  ab,5d,bd

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\Install\ADJUST\Parent Product]

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\Install\ADJUST\Parent Product\ADJUST]
"IconGroup"="ADJUST"
"IconList"=" O   A g;
W       ;       ;       ;ICONV    ;     "

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\SymfoWARE]

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\SymfoWARE\ESQL]

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\SymfoWARE\ESQL\CurrentVersion]
"Client"="C:\\SFWCLNT\\ESQL"
"InstallPath"="C:\\SFWCLNT\\ESQL"
"Lang"="japanese"

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\SymfoWARE\Setup]
"Current"="Client"

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\SymfoWARE\Setup\CLIENT]
"SrcPath"="E:\\V11L30E\\NTCL"
"RegisteredOwner"="tpc"
"RegisteredOrganization"=" x m       "
"DesPath"="C:\\SFWCLNT"
"DataPath"="C:\\SFWD"
"ScriptFile"="DUMMY"
"LogFile"=""
"InstalledDateTime"="1999/03/25 08:40:02"
"Language"="Japanese"
"Server"=dword:00000000
"Enterprise"=dword:00000000
"Version"=hex:00,20,01,01,00,00,00,00
"RELATION"="ESQLR,OLE"
"Installed"="RELATION"

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\SymfoWARE MediaService]

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\SymfoWARE MediaService\CurrentVersion]
"CurrentProduct"="SymfoWARE"
"Path"="C:\\SFWCLNT\\MEDIA"
"Version"=dword:000f6da1

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\SymfoWARE
MediaService\CurrentVersion\SymfoWARE]
"Path"="C:\\SFWCLNT\\MEDIA"
"Version"=dword:000f6da1

[HKEY_LOCAL_MACHINE\SOFTWARE\Fujitsu\TPC-C ISAPI Application]
"Term_Base"=dword:00000001
"NumWarehouses"=dword:00000ad4
"MaxUsers"=dword:00006c48
"MaxTerm of Client"=dword:000013c4
"CONTROL_Flag"=dword:00000001


[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters]
"BandwidthLevel"=dword:ffffffff
"ListenBackLog"=dword:00000019
"MemoryCacheSize"=dword:00000000
"ObjectCacheTTL"=dword:ffffffff
"PoolThreadLimit"=dword:0000020d

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters\Filter]
"FilterType"=dword:00000000
"NumGrantSites"=dword:00000000
"NumDenySites"=dword:00000000

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Parameters\MimeMap]
"text/html,htm,,h"=""
"image/gif,gif,,g"=""
"image/jpeg,jpg,,:"=""
"text/plain,txt,,0"=""
"text/html,html,,h"=""
"image/jpeg,jpeg,,:"=""
"image/jpeg,jpe,,:"=""
"image/bmp,bmp,,:"=""
"application/octet-stream,*,,5"=""
"application/pdf,pdf,,5"=""
"application/octet-stream,bin,,5"=""
"application/oda,oda,,5"=""
"application/zip,zip,,9"=""
"application/rtf,rtf,,5"=""
"application/postscript,ps,,5"=""
"application/postscript,ai,,5"=""
"application/postscript,eps,,5"=""
"application/mac-binhex40,hqx,,4"=""
"application/msword,doc,,5"=""
"application/msword,dot,,5"=""
"application/winhlp,hlp,,5"=""
"video/mpeg,mpeg,,:"=""
"video/mpeg,mpg,,:"=""
"video/mpeg,mpe,,:"=""
"video/x-msvideo,avi,,<"=""
"video/quicktime,qt,,:"=""
"video/quicktime,mov,,:"=""
"video/x-sgi-movie,movie,,<"=""
"x-world/x-vrml,wrl,,5"=""
"x-world/x-vrml,xaf,,5"=""
"x-world/x-vrml,xof,,5"=""
"x-world/x-vrml,flr,,5"=""
"x-world/x-vrml,wrz,,5"=""
"application/x-director,dcr,,5"=""
"application/x-director,dir,,5"=""
"application/x-director,dxr,,5"=""
"image/cis-cod,cod,,5"=""
"image/x-cmx,cmx,,5"=""
"application/envoy,evy,,5"=""
"application/x-msaccess,mdb,,5"=""
"application/x-mscardfile,crd,,5"=""
"application/x-msclip,clp,,5"=""
"application/octet-stream,exe,,5"=""
"application/x-msexcel,xla,,5"=""
"application/x-msexcel,xlc,,5"=""
"application/x-msexcel,xlm,,5"=""
"application/x-msexcel,xls,,5"=""
"application/x-msexcel,xlt,,5"=""
"application/x-msexcel,xlw,,5"=""
"application/x-msmediaview,m13,,5"=""
"application/x-msmediaview,m14,,5"=""
"application/x-msmoney,mny,,5"=""
"application/x-mspowerpoint,ppt,,5"=""
"application/x-msproject,mpp,,5"=""
"application/x-mspublisher,pub,,5"=""
"application/x-msterminal,trm,,5"=""
"application/x-msworks,wks,,5"=""
"application/x-mswrite,wri,,5"=""
"application/x-msmetafile,wmf,,5"=""
"application/x-csh,csh,,5"=""
"application/x-dvi,dvi,,5"=""
"application/x-hdf,hdf,,5"=""
"application/x-latex,latex,,5"=""
```

```
"application/x-netcdf,nc,,5"=""
"application/x-netcdf,cdf,,5"=""
"application/x-sh,sh,,5"=""
"application/x-tcl,tcl,,5"=""
"application/x-tex,tex,,5"=""
"application/x-texinfo,texinfo,,5"=""
"application/x-texinfo,texi,,5"=""
"application/x-troff,t,,5"=""
"application/x-troff,tr,,5"=""
"application/x-troff,roff,,5"=""
"application/x-troff-man,man,,5"=""
"application/x-troff-me,me,,5"=""
"application/x-troff-ms,ms,,5"=""
"application/x-wais-source,src,,7"=""
"application/x-bcpio,bcpio,,5"=""
"application/x-cpio,cpio,,5"=""
"application/x-gtar,gtar,,9"=""
"application/x-shar,shar,,5"=""
"application/x-sv4cpio,sv4cpio,,5"=""
"application/x-sv4crc,sv4crc,,5"=""
"application/x-tar,tar,,5"=""
"application/x-ustar,ustar,,5"=""
"audio/basic,au,,<"=""
"audio/basic,snd,,<"=""
"audio/x-aiff,aif,,<"=""
"audio/x-aiff,aiff,,<"=""
"audio/x-aiff,aifc,,<"=""
"audio/x-wav,wav,,<"=""
"audio/x-pn-realaudio,ram,,<"=""
"image/ief,ief,,:"=""
"image/tiff,tiff,,:"=""
"image/tiff,tif,,:"=""
"image/x-cmu-raster,ras,,:"=""
"image/x-portable-anymap,pnm,,:"=""
"image/x-portable-bitmap,pbm,,:"=""
"image/x-portable-graymap,pgm,,:"=""
"image/x-portable-pixmap,ppm,,:"=""
"image/x-rgb,rgb,,:"=""
"image/x-xbitmap,xbm,,:"=""
"image/x-xpixmap,xpm,,:"=""
"image/x-xwindowdump,xwd,,:"=""
"text/html,stm,,h"=""
"text/plain,bas,,0"=""
"text/plain,c,,0"=""
"text/plain,h,,0"=""
"text/richtext,rtx,,0"=""
"text/tab-separated-values,tsv,,0"=""
"text/x-setext,etx,,0"=""
"application/x-perfmon,pmc,,5"=""
"application/x-perfmon,pma,,5"=""
"application/x-perfmon,pmr,,5"=""
"application/x-perfmon,pml,,5"=""
"application/x-perfmon,pmw,,5"=""

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\InetInfo\Performance]
"Library"="infoctrs.DLL"
"Open"="OpenINFOPerformanceData"
"Close"="CloseINFOPerformanceData"
"Collect"="CollectINFOPerformanceData"
"Last Counter"=dword:00000756
"Last Help"=dword:00000757
"First Counter"=dword:00000738
"First Help"=dword:00000739


[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC]
"Type"=dword:00000020
"Start"=dword:00000002
"ErrorControl"=dword:00000000
"ImagePath"=hex(2):43,3a,5c,57,49,4e,4e,54,5c,53,79,73,74,65,6d,33,32,5c,69,6e,\
  65,74,73,72,76,5c,69,6e,65,74,69,6e,66,6f,2e,65,78,65,00
"DisplayName"="World Wide Web Publishing Service"
"DependOnService"=hex(7):52,50,43,53,53,00,4e,54,4c,4d,53,53,50,00,00
"DependOnGroup"=hex(7):00
"ObjectName"="LocalSystem"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\HTMLA]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters]
"MajorVersion"=dword:00000002
"MinorVersion"=dword:00000000
"AdminName"="Administrator"
"AdminEmail"="Admin@corp.com"
"MaxConnections"=dword:000186a0
"LogType"=dword:00000000
"LogFileDirectory"=hex(2):25,53,79,73,74,65,6d,52,6f,6f,74,25,5c,53,79,73,74,\
  65,6d,33,32,5c,4c,6f,67,46,69,6c,65,73,00
"LogFileTruncateSize"=dword:01400000
"LogFilePeriod"=dword:00000001
"LogFileFormat"=dword:00000000
"LogSqlDataSource"="HTTPLOG"
"LogSqlTableName"="Internetlog"
"LogSqlUserName"="InternetAdmin"
"LogSqlPassword"="sqllog"
"Authorization"=dword:00000005
"AnonymousUserName"="IUSR_NTCL14"
"Default Load File"="Default.htm"
"Dir Browse Control"=dword:4000001e
"CheckForWAISDB"=dword:00000000
"CacheExtensions"=dword:00000001
"GlobalExpire"=dword:ffffffff
"ServerSideIncludesEnabled"=dword:00000001
"ServerSideIncludesExtension"=".stm"
"DebugFlags"=dword:00000008
"ScriptTimeout"=dword:00000384
"ConnectionTimeOut"=dword:00000384
"InstallPath"="C:\\WINNT\\System32\\inetsrv"
"SecurePort"=dword:000001bb
"Filter DLLs"="C:\\WINNT\\System32\\inetsrv\\sspifilt.dll"
"AccessDeniedMessage"="    :                B"
"NTAuthenticationProviders"="NTLM"
"ServerComment"=""

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Script Map]
".idc"="C:\\WINNT\\System32\\inetsrv\\httpodbc.dll"


[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\Virtual Roots]
"/,"="C:\\InetPub\\wwwroot,,1"
"/tpcg,"="C:\\client\\tpapl4\\Release,,5"
"/tpc,"="C:\\client\\tpaplFMLK\\Release,,5"
"/Scripts,"="C:\\InetPub\\scripts,,4"
"/iisadmin,"="C:\\WINNT\\System32\\inetsrv\\iisadmin,,1"


[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Performance]
"Library"="w3ctrs.DLL"
"Open"="OpenW3PerformanceData"
"Close"="CloseW3PerformanceData"
"Collect"="CollectW3PerformanceData"
"Last Counter"=dword:00000790
"Last Help"=dword:00000791
"First Counter"=dword:00000758
"First Help"=dword:00000759

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Security]
"Security"=hex:01,00,14,80,c0,00,00,00,cc,00,00,00,14,00,00,00,34,00,00,00,02,\
  00,20,00,01,00,00,00,02,80,18,00,ff,01,0f,00,01,01,00,00,00,00,00,01,00,00,\
  00,00,20,02,00,00,02,00,8c,00,05,00,00,00,00,00,18,00,8d,01,02,00,01,01,00,\
  00,00,00,00,01,00,00,00,00,00,a6,14,00,00,00,1c,00,fd,01,02,00,01,02,00,00,\
  00,00,00,05,20,00,00,00,23,02,00,00,00,00,00,00,00,1c,00,ff,01,0f,00,01,\
  02,00,00,00,00,00,05,20,00,00,00,20,02,00,00,00,00,00,00,00,1c,00,ff,01,\
  0f,00,01,02,00,00,00,00,00,05,20,00,00,00,25,02,00,00,00,00,00,00,00,00,18,\
  00,fd,01,02,00,01,01,00,00,00,00,00,05,12,00,00,00,25,02,00,00,01,01,00,00,\
  00,00,00,05,12,00,00,00,01,01,00,00,00,00,00,05,12,00,00,00

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\W3SAMP]

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Enum]
"0"="Root\\LEGACY_W3SVC\\0000"
"Count"=dword:00000001
"NextInstance"=dword:00000001
```

File: conbf.sh

```
:
echo "# WORK FILE for rdbconbf" >conbf_wk1

#mkconbf2 WAREHOUSE    42  W      7  1
>>conbf_wk1
#mkconbf2 DISTRICT     42  D      7  1
>>conbf_wk1
#mkconbf2 CUSTOMER     252  C     12  6
>>conbf_wk1
#mkconbf2 CUSTOMER_X  252  C_IX  12  6
>>conbf_wk1
#mkconbf2 ORDERS       252  O     12  6
>>conbf_wk1
#mkconbf2 ORDERS_IX    252  O_IX  12  6
>>conbf_wk1
#mkconbf2 NEWORDER     252  NO    12  1
>>conbf_wk1
#mkconbf2 NEWORDER_X  252  NO_IX 12  1
>>conbf_wk1
#mkconbf2 STOCK        84  S     12  2
>>conbf_wk1
#mkconbf2 ITEM         1  I      1  1 >>conbf_wk1
```

```
#mkconbf2 ORDERLIN    252  OL    12  1
>>conbf_wk1
#mkconbf2 HISTORY     252  H     12  6
>>conbf_wk1

cp -p conbf_2772_12 conbf_wk1

timex rdbconbf -f conbf_wk1
```

File: conbf_2772_12

```
# WORK FILE for rdbconbf
TPCC.WAREHOUSE_1_DSI W_1
TPCC.WAREHOUSE_2_DSI W_2
TPCC.WAREHOUSE_3_DSI W_3
TPCC.WAREHOUSE_4_DSI W_4
TPCC.WAREHOUSE_5_DSI W_5
TPCC.WAREHOUSE_6_DSI W_6
TPCC.WAREHOUSE_7_DSI W_7
TPCC.WAREHOUSE_8_DSI W_1
TPCC.WAREHOUSE_9_DSI W_2
TPCC.WAREHOUSE_10_DSI W_3
TPCC.WAREHOUSE_11_DSI W_4
TPCC.WAREHOUSE_12_DSI W_5
TPCC.WAREHOUSE_13_DSI W_6
TPCC.WAREHOUSE_14_DSI W_7
TPCC.WAREHOUSE_15_DSI W_1
TPCC.WAREHOUSE_16_DSI W_2
TPCC.WAREHOUSE_17_DSI W_3
TPCC.WAREHOUSE_18_DSI W_4
TPCC.WAREHOUSE_19_DSI W_5
TPCC.WAREHOUSE_20_DSI W_6
TPCC.WAREHOUSE_21_DSI W_7
TPCC.WAREHOUSE_22_DSI W_1
TPCC.WAREHOUSE_23_DSI W_2
TPCC.WAREHOUSE_24_DSI W_3
TPCC.WAREHOUSE_25_DSI W_4
TPCC.WAREHOUSE_26_DSI W_5
TPCC.WAREHOUSE_27_DSI W_6
TPCC.WAREHOUSE_28_DSI W_7
TPCC.WAREHOUSE_29_DSI W_1
TPCC.WAREHOUSE_30_DSI W_2
TPCC.WAREHOUSE_31_DSI W_3
TPCC.WAREHOUSE_32_DSI W_4
TPCC.WAREHOUSE_33_DSI W_5
TPCC.WAREHOUSE_34_DSI W_6
TPCC.WAREHOUSE_35_DSI W_7
TPCC.WAREHOUSE_36_DSI W_1
TPCC.WAREHOUSE_37_DSI W_2
TPCC.WAREHOUSE_38_DSI W_3
TPCC.WAREHOUSE_39_DSI W_4
TPCC.WAREHOUSE_40_DSI W_5
TPCC.WAREHOUSE_41_DSI W_6
TPCC.WAREHOUSE_42_DSI W_7
TPCC.DISTRICT_1_DSI D_1
TPCC.DISTRICT_2_DSI D_2
TPCC.DISTRICT_3_DSI D_3
TPCC.DISTRICT_4_DSI D_4
TPCC.DISTRICT_5_DSI D_5
TPCC.DISTRICT_6_DSI D_6
TPCC.DISTRICT_7_DSI D_7
TPCC.DISTRICT_8_DSI D_1
TPCC.DISTRICT_9_DSI D_2
TPCC.DISTRICT_10_DSI D_3
TPCC.DISTRICT_11_DSI D_4
TPCC.DISTRICT_12_DSI D_5
TPCC.DISTRICT_13_DSI D_6
TPCC.DISTRICT_14_DSI D_7
TPCC.DISTRICT_15_DSI D_1
TPCC.DISTRICT_16_DSI D_2
TPCC.DISTRICT_17_DSI D_3
TPCC.DISTRICT_18_DSI D_4
TPCC.DISTRICT_19_DSI D_5
TPCC.DISTRICT_20_DSI D_6
TPCC.DISTRICT_21_DSI D_7
TPCC.DISTRICT_22_DSI D_1
TPCC.DISTRICT_23_DSI D_2
TPCC.DISTRICT_24_DSI D_3
TPCC.DISTRICT_25_DSI D_4
TPCC.DISTRICT_26_DSI D_5
TPCC.DISTRICT_27_DSI D_6
TPCC.DISTRICT_28_DSI D_7
TPCC.DISTRICT_29_DSI D_1
TPCC.DISTRICT_30_DSI D_2
TPCC.DISTRICT_31_DSI D_3
TPCC.DISTRICT_32_DSI D_4
TPCC.DISTRICT_33_DSI D_5
TPCC.DISTRICT_34_DSI D_6
TPCC.DISTRICT_35_DSI D_7
TPCC.DISTRICT_36_DSI D_1
TPCC.DISTRICT_37_DSI D_2
TPCC.DISTRICT_38_DSI D_3
TPCC.DISTRICT_39_DSI D_4
TPCC.DISTRICT_40_DSI D_5
TPCC.DISTRICT_41_DSI D_6
TPCC.DISTRICT_42_DSI D_7
TPCC.CUSTOMER_1_DSI C_1
TPCC.CUSTOMER_2_DSI C_1
TPCC.CUSTOMER_3_DSI C_1
TPCC.CUSTOMER_4_DSI C_1
TPCC.CUSTOMER_5_DSI C_1
TPCC.CUSTOMER_6_DSI C_1
TPCC.CUSTOMER_7_DSI C_2
TPCC.CUSTOMER_8_DSI C_2
TPCC.CUSTOMER_9_DSI C_2
TPCC.CUSTOMER_10_DSI C_2
TPCC.CUSTOMER_11_DSI C_2
TPCC.CUSTOMER_12_DSI C_2
TPCC.CUSTOMER_13_DSI C_3
TPCC.CUSTOMER_14_DSI C_3
TPCC.CUSTOMER_15_DSI C_3
TPCC.CUSTOMER_16_DSI C_3
TPCC.CUSTOMER_17_DSI C_3
TPCC.CUSTOMER_18_DSI C_3
TPCC.CUSTOMER_19_DSI C_4
TPCC.CUSTOMER_20_DSI C_4
TPCC.CUSTOMER_21_DSI C_4
TPCC.CUSTOMER_22_DSI C_4
TPCC.CUSTOMER_23_DSI C_4
TPCC.CUSTOMER_24_DSI C_4
TPCC.CUSTOMER_25_DSI C_5
TPCC.CUSTOMER_26_DSI C_5
TPCC.CUSTOMER_27_DSI C_5
TPCC.CUSTOMER_28_DSI C_5
TPCC.CUSTOMER_29_DSI C_5
TPCC.CUSTOMER_30_DSI C_5
TPCC.CUSTOMER_31_DSI C_6
TPCC.CUSTOMER_32_DSI C_6
TPCC.CUSTOMER_33_DSI C_6
TPCC.CUSTOMER_34_DSI C_6
TPCC.CUSTOMER_35_DSI C_6
TPCC.CUSTOMER_36_DSI C_6
TPCC.CUSTOMER_37_DSI C_7
TPCC.CUSTOMER_38_DSI C_7
TPCC.CUSTOMER_39_DSI C_7
TPCC.CUSTOMER_40_DSI C_7
TPCC.CUSTOMER_41_DSI C_7
TPCC.CUSTOMER_42_DSI C_7
TPCC.CUSTOMER_43_DSI C_8
TPCC.CUSTOMER_44_DSI C_8
TPCC.CUSTOMER_45_DSI C_8
TPCC.CUSTOMER_46_DSI C_8
TPCC.CUSTOMER_47_DSI C_8
TPCC.CUSTOMER_48_DSI C_8
TPCC.CUSTOMER_49_DSI C_9
TPCC.CUSTOMER_50_DSI C_9
TPCC.CUSTOMER_51_DSI C_9
TPCC.CUSTOMER_52_DSI C_9
TPCC.CUSTOMER_53_DSI C_9
TPCC.CUSTOMER_54_DSI C_9
TPCC.CUSTOMER_55_DSI C_10
TPCC.CUSTOMER_56_DSI C_10
TPCC.CUSTOMER_57_DSI C_10
TPCC.CUSTOMER_58_DSI C_10
TPCC.CUSTOMER_59_DSI C_10
TPCC.CUSTOMER_60_DSI C_10
TPCC.CUSTOMER_61_DSI C_11
TPCC.CUSTOMER_62_DSI C_11
TPCC.CUSTOMER_63_DSI C_11
TPCC.CUSTOMER_64_DSI C_11
TPCC.CUSTOMER_65_DSI C_11
TPCC.CUSTOMER_66_DSI C_11
TPCC.CUSTOMER_67_DSI C_12
TPCC.CUSTOMER_68_DSI C_12
TPCC.CUSTOMER_69_DSI C_12
TPCC.CUSTOMER_70_DSI C_12
TPCC.CUSTOMER_71_DSI C_12
TPCC.CUSTOMER_72_DSI C_12
TPCC.CUSTOMER_73_DSI C_1
TPCC.CUSTOMER_74_DSI C_1
TPCC.CUSTOMER_75_DSI C_1
TPCC.CUSTOMER_76_DSI C_1
TPCC.CUSTOMER_77_DSI C_1
TPCC.CUSTOMER_78_DSI C_1
TPCC.CUSTOMER_79_DSI C_2
TPCC.CUSTOMER_80_DSI C_2
TPCC.CUSTOMER_81_DSI C_2
TPCC.CUSTOMER_82_DSI C_2
TPCC.CUSTOMER_83_DSI C_2
TPCC.CUSTOMER_84_DSI C_2
TPCC.CUSTOMER_85_DSI C_3
TPCC.CUSTOMER_86_DSI C_3
TPCC.CUSTOMER_87_DSI C_3
TPCC.CUSTOMER_88_DSI C_3
TPCC.CUSTOMER_89_DSI C_3
TPCC.CUSTOMER_90_DSI C_3
TPCC.CUSTOMER_91_DSI C_4
TPCC.CUSTOMER_92_DSI C_4
TPCC.CUSTOMER_93_DSI C_4
TPCC.CUSTOMER_94_DSI C_4
TPCC.CUSTOMER_95_DSI C_4
TPCC.CUSTOMER_96_DSI C_4
TPCC.CUSTOMER_97_DSI C_5
TPCC.CUSTOMER_98_DSI C_5
TPCC.CUSTOMER_99_DSI C_5
TPCC.CUSTOMER_100_DSI C_5
TPCC.CUSTOMER_101_DSI C_5
TPCC.CUSTOMER_102_DSI C_5
TPCC.CUSTOMER_103_DSI C_6
TPCC.CUSTOMER_104_DSI C_6
TPCC.CUSTOMER_105_DSI C_6
```

TPCC.CUSTOMER_106_DSI C_6
TPCC.CUSTOMER_107_DSI C_6
TPCC.CUSTOMER_108_DSI C_6
TPCC.CUSTOMER_109_DSI C_7
TPCC.CUSTOMER_110_DSI C_7
TPCC.CUSTOMER_111_DSI C_7
TPCC.CUSTOMER_112_DSI C_7
TPCC.CUSTOMER_113_DSI C_7
TPCC.CUSTOMER_114_DSI C_7
TPCC.CUSTOMER_115_DSI C_8
TPCC.CUSTOMER_116_DSI C_8
TPCC.CUSTOMER_117_DSI C_8
TPCC.CUSTOMER_118_DSI C_8
TPCC.CUSTOMER_119_DSI C_8
TPCC.CUSTOMER_120_DSI C_8
TPCC.CUSTOMER_121_DSI C_9
TPCC.CUSTOMER_122_DSI C_9
TPCC.CUSTOMER_123_DSI C_9
TPCC.CUSTOMER_124_DSI C_9
TPCC.CUSTOMER_125_DSI C_9
TPCC.CUSTOMER_126_DSI C_9
TPCC.CUSTOMER_127_DSI C_10
TPCC.CUSTOMER_128_DSI C_10
TPCC.CUSTOMER_129_DSI C_10
TPCC.CUSTOMER_130_DSI C_10
TPCC.CUSTOMER_131_DSI C_10
TPCC.CUSTOMER_132_DSI C_10
TPCC.CUSTOMER_133_DSI C_11
TPCC.CUSTOMER_134_DSI C_11
TPCC.CUSTOMER_135_DSI C_11
TPCC.CUSTOMER_136_DSI C_11
TPCC.CUSTOMER_137_DSI C_11
TPCC.CUSTOMER_138_DSI C_11
TPCC.CUSTOMER_139_DSI C_12
TPCC.CUSTOMER_140_DSI C_12
TPCC.CUSTOMER_141_DSI C_12
TPCC.CUSTOMER_142_DSI C_12
TPCC.CUSTOMER_143_DSI C_12
TPCC.CUSTOMER_144_DSI C_12
TPCC.CUSTOMER_145_DSI C_1
TPCC.CUSTOMER_146_DSI C_1
TPCC.CUSTOMER_147_DSI C_1
TPCC.CUSTOMER_148_DSI C_1
TPCC.CUSTOMER_149_DSI C_1
TPCC.CUSTOMER_150_DSI C_1
TPCC.CUSTOMER_151_DSI C_2
TPCC.CUSTOMER_152_DSI C_2
TPCC.CUSTOMER_153_DSI C_2
TPCC.CUSTOMER_154_DSI C_2
TPCC.CUSTOMER_155_DSI C_2
TPCC.CUSTOMER_156_DSI C_2
TPCC.CUSTOMER_157_DSI C_3
TPCC.CUSTOMER_158_DSI C_3
TPCC.CUSTOMER_159_DSI C_3
TPCC.CUSTOMER_160_DSI C_3
TPCC.CUSTOMER_161_DSI C_3
TPCC.CUSTOMER_162_DSI C_3
TPCC.CUSTOMER_163_DSI C_4
TPCC.CUSTOMER_164_DSI C_4
TPCC.CUSTOMER_165_DSI C_4
TPCC.CUSTOMER_166_DSI C_4
TPCC.CUSTOMER_167_DSI C_4
TPCC.CUSTOMER_168_DSI C_4
TPCC.CUSTOMER_169_DSI C_5
TPCC.CUSTOMER_170_DSI C_5
TPCC.CUSTOMER_171_DSI C_5
TPCC.CUSTOMER_172_DSI C_5
TPCC.CUSTOMER_173_DSI C_5
TPCC.CUSTOMER_174_DSI C_5
TPCC.CUSTOMER_175_DSI C_6
TPCC.CUSTOMER_176_DSI C_6
TPCC.CUSTOMER_177_DSI C_6
TPCC.CUSTOMER_178_DSI C_6
TPCC.CUSTOMER_179_DSI C_6
TPCC.CUSTOMER_180_DSI C_6
TPCC.CUSTOMER_181_DSI C_7
TPCC.CUSTOMER_182_DSI C_7
TPCC.CUSTOMER_183_DSI C_7
TPCC.CUSTOMER_184_DSI C_7
TPCC.CUSTOMER_185_DSI C_7
TPCC.CUSTOMER_186_DSI C_7
TPCC.CUSTOMER_187_DSI C_8
TPCC.CUSTOMER_188_DSI C_8
TPCC.CUSTOMER_189_DSI C_8
TPCC.CUSTOMER_190_DSI C_8
TPCC.CUSTOMER_191_DSI C_8
TPCC.CUSTOMER_192_DSI C_8
TPCC.CUSTOMER_193_DSI C_9
TPCC.CUSTOMER_194_DSI C_9
TPCC.CUSTOMER_195_DSI C_9
TPCC.CUSTOMER_196_DSI C_9
TPCC.CUSTOMER_197_DSI C_9
TPCC.CUSTOMER_198_DSI C_9
TPCC.CUSTOMER_199_DSI C_10
TPCC.CUSTOMER_200_DSI C_10
TPCC.CUSTOMER_201_DSI C_10
TPCC.CUSTOMER_202_DSI C_10
TPCC.CUSTOMER_203_DSI C_10
TPCC.CUSTOMER_204_DSI C_10
TPCC.CUSTOMER_205_DSI C_11
TPCC.CUSTOMER_206_DSI C_11
TPCC.CUSTOMER_207_DSI C_11
TPCC.CUSTOMER_208_DSI C_11
TPCC.CUSTOMER_209_DSI C_11
TPCC.CUSTOMER_210_DSI C_11
TPCC.CUSTOMER_211_DSI C_12
TPCC.CUSTOMER_212_DSI C_12
TPCC.CUSTOMER_213_DSI C_12
TPCC.CUSTOMER_214_DSI C_12
TPCC.CUSTOMER_215_DSI C_12
TPCC.CUSTOMER_216_DSI C_12
TPCC.CUSTOMER_217_DSI C_1
TPCC.CUSTOMER_218_DSI C_1
TPCC.CUSTOMER_219_DSI C_1
TPCC.CUSTOMER_220_DSI C_1
TPCC.CUSTOMER_221_DSI C_1
TPCC.CUSTOMER_222_DSI C_1
TPCC.CUSTOMER_223_DSI C_2
TPCC.CUSTOMER_224_DSI C_2
TPCC.CUSTOMER_225_DSI C_2
TPCC.CUSTOMER_226_DSI C_2
TPCC.CUSTOMER_227_DSI C_2
TPCC.CUSTOMER_228_DSI C_2
TPCC.CUSTOMER_229_DSI C_7
TPCC.CUSTOMER_230_DSI C_7
TPCC.CUSTOMER_231_DSI C_8
TPCC.CUSTOMER_232_DSI C_8
TPCC.CUSTOMER_233_DSI C_9
TPCC.CUSTOMER_234_DSI C_9
TPCC.CUSTOMER_235_DSI C_10
TPCC.CUSTOMER_236_DSI C_10
TPCC.CUSTOMER_237_DSI C_5
TPCC.CUSTOMER_238_DSI C_5
TPCC.CUSTOMER_239_DSI C_6
TPCC.CUSTOMER_240_DSI C_6
TPCC.CUSTOMER_241_DSI C_3
TPCC.CUSTOMER_242_DSI C_3
TPCC.CUSTOMER_243_DSI C_4
TPCC.CUSTOMER_244_DSI C_4
TPCC.CUSTOMER_245_DSI C_5
TPCC.CUSTOMER_246_DSI C_5
TPCC.CUSTOMER_247_DSI C_6
TPCC.CUSTOMER_248_DSI C_6
TPCC.CUSTOMER_249_DSI C_1
TPCC.CUSTOMER_250_DSI C_1
TPCC.CUSTOMER_251_DSI C_2
TPCC.CUSTOMER_252_DSI C_2
TPCC.CUSTOMER_X_1_DSI C_IX_1
TPCC.CUSTOMER_X_2_DSI C_IX_1
TPCC.CUSTOMER_X_3_DSI C_IX_1
TPCC.CUSTOMER_X_4_DSI C_IX_1
TPCC.CUSTOMER_X_5_DSI C_IX_1
TPCC.CUSTOMER_X_6_DSI C_IX_1
TPCC.CUSTOMER_X_7_DSI C_IX_2
TPCC.CUSTOMER_X_8_DSI C_IX_2
TPCC.CUSTOMER_X_9_DSI C_IX_2
TPCC.CUSTOMER_X_10_DSI C_IX_2
TPCC.CUSTOMER_X_11_DSI C_IX_2
TPCC.CUSTOMER_X_12_DSI C_IX_2
TPCC.CUSTOMER_X_13_DSI C_IX_3
TPCC.CUSTOMER_X_14_DSI C_IX_3
TPCC.CUSTOMER_X_15_DSI C_IX_3
TPCC.CUSTOMER_X_16_DSI C_IX_3
TPCC.CUSTOMER_X_17_DSI C_IX_3
TPCC.CUSTOMER_X_18_DSI C_IX_3
TPCC.CUSTOMER_X_19_DSI C_IX_4
TPCC.CUSTOMER_X_20_DSI C_IX_4
TPCC.CUSTOMER_X_21_DSI C_IX_4
TPCC.CUSTOMER_X_22_DSI C_IX_4
TPCC.CUSTOMER_X_23_DSI C_IX_4
TPCC.CUSTOMER_X_24_DSI C_IX_4
TPCC.CUSTOMER_X_25_DSI C_IX_5
TPCC.CUSTOMER_X_26_DSI C_IX_5
TPCC.CUSTOMER_X_27_DSI C_IX_5
TPCC.CUSTOMER_X_28_DSI C_IX_5
TPCC.CUSTOMER_X_29_DSI C_IX_5
TPCC.CUSTOMER_X_30_DSI C_IX_5
TPCC.CUSTOMER_X_31_DSI C_IX_6
TPCC.CUSTOMER_X_32_DSI C_IX_6
TPCC.CUSTOMER_X_33_DSI C_IX_6
TPCC.CUSTOMER_X_34_DSI C_IX_6
TPCC.CUSTOMER_X_35_DSI C_IX_6
TPCC.CUSTOMER_X_36_DSI C_IX_6
TPCC.CUSTOMER_X_37_DSI C_IX_7
TPCC.CUSTOMER_X_38_DSI C_IX_7
TPCC.CUSTOMER_X_39_DSI C_IX_7
TPCC.CUSTOMER_X_40_DSI C_IX_7
TPCC.CUSTOMER_X_41_DSI C_IX_7
TPCC.CUSTOMER_X_42_DSI C_IX_7
TPCC.CUSTOMER_X_43_DSI C_IX_8
TPCC.CUSTOMER_X_44_DSI C_IX_8
TPCC.CUSTOMER_X_45_DSI C_IX_8
TPCC.CUSTOMER_X_46_DSI C_IX_8
TPCC.CUSTOMER_X_47_DSI C_IX_8
TPCC.CUSTOMER_X_48_DSI C_IX_8
TPCC.CUSTOMER_X_49_DSI C_IX_9
TPCC.CUSTOMER_X_50_DSI C_IX_9
TPCC.CUSTOMER_X_51_DSI C_IX_9
TPCC.CUSTOMER_X_52_DSI C_IX_9
TPCC.CUSTOMER_X_53_DSI C_IX_9
TPCC.CUSTOMER_X_54_DSI C_IX_9
TPCC.CUSTOMER_X_55_DSI C_IX_10
TPCC.CUSTOMER_X_56_DSI C_IX_10
TPCC.CUSTOMER_X_57_DSI C_IX_10

TPCC.CUSTOMER_X_58_DSI C_IX_10
TPCC.CUSTOMER_X_59_DSI C_IX_10
TPCC.CUSTOMER_X_60_DSI C_IX_10
TPCC.CUSTOMER_X_61_DSI C_IX_11
TPCC.CUSTOMER_X_62_DSI C_IX_11
TPCC.CUSTOMER_X_63_DSI C_IX_11
TPCC.CUSTOMER_X_64_DSI C_IX_11
TPCC.CUSTOMER_X_65_DSI C_IX_11
TPCC.CUSTOMER_X_66_DSI C_IX_11
TPCC.CUSTOMER_X_67_DSI C_IX_12
TPCC.CUSTOMER_X_68_DSI C_IX_12
TPCC.CUSTOMER_X_69_DSI C_IX_12
TPCC.CUSTOMER_X_70_DSI C_IX_12
TPCC.CUSTOMER_X_71_DSI C_IX_12
TPCC.CUSTOMER_X_72_DSI C_IX_12
TPCC.CUSTOMER_X_73_DSI C_IX_1
TPCC.CUSTOMER_X_74_DSI C_IX_1
TPCC.CUSTOMER_X_75_DSI C_IX_1
TPCC.CUSTOMER_X_76_DSI C_IX_1
TPCC.CUSTOMER_X_77_DSI C_IX_1
TPCC.CUSTOMER_X_78_DSI C_IX_1
TPCC.CUSTOMER_X_79_DSI C_IX_2
TPCC.CUSTOMER_X_80_DSI C_IX_2
TPCC.CUSTOMER_X_81_DSI C_IX_2
TPCC.CUSTOMER_X_82_DSI C_IX_2
TPCC.CUSTOMER_X_83_DSI C_IX_2
TPCC.CUSTOMER_X_84_DSI C_IX_2
TPCC.CUSTOMER_X_85_DSI C_IX_3
TPCC.CUSTOMER_X_86_DSI C_IX_3
TPCC.CUSTOMER_X_87_DSI C_IX_3
TPCC.CUSTOMER_X_88_DSI C_IX_3
TPCC.CUSTOMER_X_89_DSI C_IX_3
TPCC.CUSTOMER_X_90_DSI C_IX_3
TPCC.CUSTOMER_X_91_DSI C_IX_4
TPCC.CUSTOMER_X_92_DSI C_IX_4
TPCC.CUSTOMER_X_93_DSI C_IX_4
TPCC.CUSTOMER_X_94_DSI C_IX_4
TPCC.CUSTOMER_X_95_DSI C_IX_4
TPCC.CUSTOMER_X_96_DSI C_IX_4
TPCC.CUSTOMER_X_97_DSI C_IX_5
TPCC.CUSTOMER_X_98_DSI C_IX_5
TPCC.CUSTOMER_X_99_DSI C_IX_5
TPCC.CUSTOMER_X_100_DSI C_IX_5
TPCC.CUSTOMER_X_101_DSI C_IX_5
TPCC.CUSTOMER_X_102_DSI C_IX_5
TPCC.CUSTOMER_X_103_DSI C_IX_6
TPCC.CUSTOMER_X_104_DSI C_IX_6
TPCC.CUSTOMER_X_105_DSI C_IX_6
TPCC.CUSTOMER_X_106_DSI C_IX_6
TPCC.CUSTOMER_X_107_DSI C_IX_6
TPCC.CUSTOMER_X_108_DSI C_IX_6
TPCC.CUSTOMER_X_109_DSI C_IX_7
TPCC.CUSTOMER_X_110_DSI C_IX_7
TPCC.CUSTOMER_X_111_DSI C_IX_7
TPCC.CUSTOMER_X_112_DSI C_IX_7
TPCC.CUSTOMER_X_113_DSI C_IX_7
TPCC.CUSTOMER_X_114_DSI C_IX_7
TPCC.CUSTOMER_X_115_DSI C_IX_8
TPCC.CUSTOMER_X_116_DSI C_IX_8
TPCC.CUSTOMER_X_117_DSI C_IX_8
TPCC.CUSTOMER_X_118_DSI C_IX_8
TPCC.CUSTOMER_X_119_DSI C_IX_8
TPCC.CUSTOMER_X_120_DSI C_IX_8
TPCC.CUSTOMER_X_121_DSI C_IX_9
TPCC.CUSTOMER_X_122_DSI C_IX_9
TPCC.CUSTOMER_X_123_DSI C_IX_9
TPCC.CUSTOMER_X_124_DSI C_IX_9
TPCC.CUSTOMER_X_125_DSI C_IX_9
TPCC.CUSTOMER_X_126_DSI C_IX_9
TPCC.CUSTOMER_X_127_DSI C_IX_10
TPCC.CUSTOMER_X_128_DSI C_IX_10
TPCC.CUSTOMER_X_129_DSI C_IX_10
TPCC.CUSTOMER_X_130_DSI C_IX_10
TPCC.CUSTOMER_X_131_DSI C_IX_10
TPCC.CUSTOMER_X_132_DSI C_IX_10
TPCC.CUSTOMER_X_133_DSI C_IX_11
TPCC.CUSTOMER_X_134_DSI C_IX_11
TPCC.CUSTOMER_X_135_DSI C_IX_11
TPCC.CUSTOMER_X_136_DSI C_IX_11
TPCC.CUSTOMER_X_137_DSI C_IX_11
TPCC.CUSTOMER_X_138_DSI C_IX_11
TPCC.CUSTOMER_X_139_DSI C_IX_12
TPCC.CUSTOMER_X_140_DSI C_IX_12
TPCC.CUSTOMER_X_141_DSI C_IX_12
TPCC.CUSTOMER_X_142_DSI C_IX_12
TPCC.CUSTOMER_X_143_DSI C_IX_12
TPCC.CUSTOMER_X_144_DSI C_IX_12
TPCC.CUSTOMER_X_145_DSI C_IX_1
TPCC.CUSTOMER_X_146_DSI C_IX_1
TPCC.CUSTOMER_X_147_DSI C_IX_1
TPCC.CUSTOMER_X_148_DSI C_IX_1
TPCC.CUSTOMER_X_149_DSI C_IX_1
TPCC.CUSTOMER_X_150_DSI C_IX_1
TPCC.CUSTOMER_X_151_DSI C_IX_2
TPCC.CUSTOMER_X_152_DSI C_IX_2
TPCC.CUSTOMER_X_153_DSI C_IX_2
TPCC.CUSTOMER_X_154_DSI C_IX_2
TPCC.CUSTOMER_X_155_DSI C_IX_2
TPCC.CUSTOMER_X_156_DSI C_IX_2
TPCC.CUSTOMER_X_157_DSI C_IX_3
TPCC.CUSTOMER_X_158_DSI C_IX_3
TPCC.CUSTOMER_X_159_DSI C_IX_3
TPCC.CUSTOMER_X_160_DSI C_IX_3
TPCC.CUSTOMER_X_161_DSI C_IX_3
TPCC.CUSTOMER_X_162_DSI C_IX_3
TPCC.CUSTOMER_X_163_DSI C_IX_4
TPCC.CUSTOMER_X_164_DSI C_IX_4
TPCC.CUSTOMER_X_165_DSI C_IX_4
TPCC.CUSTOMER_X_166_DSI C_IX_4
TPCC.CUSTOMER_X_167_DSI C_IX_4
TPCC.CUSTOMER_X_168_DSI C_IX_4
TPCC.CUSTOMER_X_169_DSI C_IX_5
TPCC.CUSTOMER_X_170_DSI C_IX_5
TPCC.CUSTOMER_X_171_DSI C_IX_5
TPCC.CUSTOMER_X_172_DSI C_IX_5
TPCC.CUSTOMER_X_173_DSI C_IX_5
TPCC.CUSTOMER_X_174_DSI C_IX_5
TPCC.CUSTOMER_X_175_DSI C_IX_6
TPCC.CUSTOMER_X_176_DSI C_IX_6
TPCC.CUSTOMER_X_177_DSI C_IX_6
TPCC.CUSTOMER_X_178_DSI C_IX_6
TPCC.CUSTOMER_X_179_DSI C_IX_6
TPCC.CUSTOMER_X_180_DSI C_IX_6
TPCC.CUSTOMER_X_181_DSI C_IX_7
TPCC.CUSTOMER_X_182_DSI C_IX_7
TPCC.CUSTOMER_X_183_DSI C_IX_7
TPCC.CUSTOMER_X_184_DSI C_IX_7
TPCC.CUSTOMER_X_185_DSI C_IX_7
TPCC.CUSTOMER_X_186_DSI C_IX_7
TPCC.CUSTOMER_X_187_DSI C_IX_8
TPCC.CUSTOMER_X_188_DSI C_IX_8
TPCC.CUSTOMER_X_189_DSI C_IX_8
TPCC.CUSTOMER_X_190_DSI C_IX_8
TPCC.CUSTOMER_X_191_DSI C_IX_8
TPCC.CUSTOMER_X_192_DSI C_IX_8
TPCC.CUSTOMER_X_193_DSI C_IX_9
TPCC.CUSTOMER_X_194_DSI C_IX_9
TPCC.CUSTOMER_X_195_DSI C_IX_9
TPCC.CUSTOMER_X_196_DSI C_IX_9
TPCC.CUSTOMER_X_197_DSI C_IX_9
TPCC.CUSTOMER_X_198_DSI C_IX_9
TPCC.CUSTOMER_X_199_DSI C_IX_10
TPCC.CUSTOMER_X_200_DSI C_IX_10
TPCC.CUSTOMER_X_201_DSI C_IX_10
TPCC.CUSTOMER_X_202_DSI C_IX_10
TPCC.CUSTOMER_X_203_DSI C_IX_10
TPCC.CUSTOMER_X_204_DSI C_IX_10
TPCC.CUSTOMER_X_205_DSI C_IX_11
TPCC.CUSTOMER_X_206_DSI C_IX_11
TPCC.CUSTOMER_X_207_DSI C_IX_11
TPCC.CUSTOMER_X_208_DSI C_IX_11
TPCC.CUSTOMER_X_209_DSI C_IX_11
TPCC.CUSTOMER_X_210_DSI C_IX_11
TPCC.CUSTOMER_X_211_DSI C_IX_12
TPCC.CUSTOMER_X_212_DSI C_IX_12
TPCC.CUSTOMER_X_213_DSI C_IX_12
TPCC.CUSTOMER_X_214_DSI C_IX_12
TPCC.CUSTOMER_X_215_DSI C_IX_12
TPCC.CUSTOMER_X_216_DSI C_IX_12
TPCC.CUSTOMER_X_217_DSI C_IX_1
TPCC.CUSTOMER_X_218_DSI C_IX_1
TPCC.CUSTOMER_X_219_DSI C_IX_1
TPCC.CUSTOMER_X_220_DSI C_IX_1
TPCC.CUSTOMER_X_221_DSI C_IX_1
TPCC.CUSTOMER_X_222_DSI C_IX_1
TPCC.CUSTOMER_X_223_DSI C_IX_2
TPCC.CUSTOMER_X_224_DSI C_IX_2
TPCC.CUSTOMER_X_225_DSI C_IX_2
TPCC.CUSTOMER_X_226_DSI C_IX_2
TPCC.CUSTOMER_X_227_DSI C_IX_2
TPCC.CUSTOMER_X_228_DSI C_IX_2
TPCC.CUSTOMER_X_229_DSI C_IX_7
TPCC.CUSTOMER_X_230_DSI C_IX_7
TPCC.CUSTOMER_X_231_DSI C_IX_8
TPCC.CUSTOMER_X_232_DSI C_IX_8
TPCC.CUSTOMER_X_233_DSI C_IX_9
TPCC.CUSTOMER_X_234_DSI C_IX_9
TPCC.CUSTOMER_X_235_DSI C_IX_10
TPCC.CUSTOMER_X_236_DSI C_IX_10
TPCC.CUSTOMER_X_237_DSI C_IX_5
TPCC.CUSTOMER_X_238_DSI C_IX_5
TPCC.CUSTOMER_X_239_DSI C_IX_6
TPCC.CUSTOMER_X_240_DSI C_IX_6
TPCC.CUSTOMER_X_241_DSI C_IX_3
TPCC.CUSTOMER_X_242_DSI C_IX_3
TPCC.CUSTOMER_X_243_DSI C_IX_4
TPCC.CUSTOMER_X_244_DSI C_IX_4
TPCC.CUSTOMER_X_245_DSI C_IX_5
TPCC.CUSTOMER_X_246_DSI C_IX_5
TPCC.CUSTOMER_X_247_DSI C_IX_6
TPCC.CUSTOMER_X_248_DSI C_IX_6
TPCC.CUSTOMER_X_249_DSI C_IX_1
TPCC.CUSTOMER_X_250_DSI C_IX_1
TPCC.CUSTOMER_X_251_DSI C_IX_2
TPCC.CUSTOMER_X_252_DSI C_IX_2
TPCC.ORDERS_1_DSI O_1
TPCC.ORDERS_2_DSI O_1
TPCC.ORDERS_3_DSI O_1
TPCC.ORDERS_4_DSI O_1
TPCC.ORDERS_5_DSI O_1
TPCC.ORDERS_6_DSI O_1
TPCC.ORDERS_7_DSI O_2
TPCC.ORDERS_8_DSI O_2
TPCC.ORDERS_9_DSI O_2

TPCC.ORDERS_10_DSI O_2
TPCC.ORDERS_11_DSI O_2
TPCC.ORDERS_12_DSI O_2
TPCC.ORDERS_13_DSI O_3
TPCC.ORDERS_14_DSI O_3
TPCC.ORDERS_15_DSI O_3
TPCC.ORDERS_16_DSI O_3
TPCC.ORDERS_17_DSI O_3
TPCC.ORDERS_18_DSI O_3
TPCC.ORDERS_19_DSI O_4
TPCC.ORDERS_20_DSI O_4
TPCC.ORDERS_21_DSI O_4
TPCC.ORDERS_22_DSI O_4
TPCC.ORDERS_23_DSI O_4
TPCC.ORDERS_24_DSI O_4
TPCC.ORDERS_25_DSI O_5
TPCC.ORDERS_26_DSI O_5
TPCC.ORDERS_27_DSI O_5
TPCC.ORDERS_28_DSI O_5
TPCC.ORDERS_29_DSI O_5
TPCC.ORDERS_30_DSI O_5
TPCC.ORDERS_31_DSI O_6
TPCC.ORDERS_32_DSI O_6
TPCC.ORDERS_33_DSI O_6
TPCC.ORDERS_34_DSI O_6
TPCC.ORDERS_35_DSI O_6
TPCC.ORDERS_36_DSI O_6
TPCC.ORDERS_37_DSI O_7
TPCC.ORDERS_38_DSI O_7
TPCC.ORDERS_39_DSI O_7
TPCC.ORDERS_40_DSI O_7
TPCC.ORDERS_41_DSI O_7
TPCC.ORDERS_42_DSI O_7
TPCC.ORDERS_43_DSI O_8
TPCC.ORDERS_44_DSI O_8
TPCC.ORDERS_45_DSI O_8
TPCC.ORDERS_46_DSI O_8
TPCC.ORDERS_47_DSI O_8
TPCC.ORDERS_48_DSI O_8
TPCC.ORDERS_49_DSI O_9
TPCC.ORDERS_50_DSI O_9
TPCC.ORDERS_51_DSI O_9
TPCC.ORDERS_52_DSI O_9
TPCC.ORDERS_53_DSI O_9
TPCC.ORDERS_54_DSI O_9
TPCC.ORDERS_55_DSI O_10
TPCC.ORDERS_56_DSI O_10
TPCC.ORDERS_57_DSI O_10
TPCC.ORDERS_58_DSI O_10
TPCC.ORDERS_59_DSI O_10
TPCC.ORDERS_60_DSI O_10
TPCC.ORDERS_61_DSI O_11
TPCC.ORDERS_62_DSI O_11
TPCC.ORDERS_63_DSI O_11
TPCC.ORDERS_64_DSI O_11
TPCC.ORDERS_65_DSI O_11
TPCC.ORDERS_66_DSI O_11
TPCC.ORDERS_67_DSI O_12
TPCC.ORDERS_68_DSI O_12
TPCC.ORDERS_69_DSI O_12
TPCC.ORDERS_70_DSI O_12
TPCC.ORDERS_71_DSI O_12
TPCC.ORDERS_72_DSI O_12
TPCC.ORDERS_73_DSI O_1
TPCC.ORDERS_74_DSI O_1
TPCC.ORDERS_75_DSI O_1
TPCC.ORDERS_76_DSI O_1
TPCC.ORDERS_77_DSI O_1
TPCC.ORDERS_78_DSI O_1
TPCC.ORDERS_79_DSI O_2
TPCC.ORDERS_80_DSI O_2
TPCC.ORDERS_81_DSI O_2
TPCC.ORDERS_82_DSI O_2
TPCC.ORDERS_83_DSI O_2
TPCC.ORDERS_84_DSI O_2
TPCC.ORDERS_85_DSI O_3
TPCC.ORDERS_86_DSI O_3
TPCC.ORDERS_87_DSI O_3
TPCC.ORDERS_88_DSI O_3
TPCC.ORDERS_89_DSI O_3
TPCC.ORDERS_90_DSI O_3
TPCC.ORDERS_91_DSI O_4
TPCC.ORDERS_92_DSI O_4
TPCC.ORDERS_93_DSI O_4
TPCC.ORDERS_94_DSI O_4
TPCC.ORDERS_95_DSI O_4
TPCC.ORDERS_96_DSI O_4
TPCC.ORDERS_97_DSI O_5
TPCC.ORDERS_98_DSI O_5
TPCC.ORDERS_99_DSI O_5
TPCC.ORDERS_100_DSI O_5
TPCC.ORDERS_101_DSI O_5
TPCC.ORDERS_102_DSI O_5
TPCC.ORDERS_103_DSI O_6
TPCC.ORDERS_104_DSI O_6
TPCC.ORDERS_105_DSI O_6
TPCC.ORDERS_106_DSI O_6
TPCC.ORDERS_107_DSI O_6
TPCC.ORDERS_108_DSI O_6
TPCC.ORDERS_109_DSI O_7
TPCC.ORDERS_110_DSI O_7
TPCC.ORDERS_111_DSI O_7
TPCC.ORDERS_112_DSI O_7
TPCC.ORDERS_113_DSI O_7
TPCC.ORDERS_114_DSI O_7
TPCC.ORDERS_115_DSI O_8
TPCC.ORDERS_116_DSI O_8
TPCC.ORDERS_117_DSI O_8
TPCC.ORDERS_118_DSI O_8
TPCC.ORDERS_119_DSI O_8
TPCC.ORDERS_120_DSI O_8
TPCC.ORDERS_121_DSI O_9
TPCC.ORDERS_122_DSI O_9
TPCC.ORDERS_123_DSI O_9
TPCC.ORDERS_124_DSI O_9
TPCC.ORDERS_125_DSI O_9
TPCC.ORDERS_126_DSI O_9
TPCC.ORDERS_127_DSI O_10
TPCC.ORDERS_128_DSI O_10
TPCC.ORDERS_129_DSI O_10
TPCC.ORDERS_130_DSI O_10
TPCC.ORDERS_131_DSI O_10
TPCC.ORDERS_132_DSI O_10
TPCC.ORDERS_133_DSI O_11
TPCC.ORDERS_134_DSI O_11
TPCC.ORDERS_135_DSI O_11
TPCC.ORDERS_136_DSI O_11
TPCC.ORDERS_137_DSI O_11
TPCC.ORDERS_138_DSI O_11
TPCC.ORDERS_139_DSI O_12
TPCC.ORDERS_140_DSI O_12
TPCC.ORDERS_141_DSI O_12
TPCC.ORDERS_142_DSI O_12
TPCC.ORDERS_143_DSI O_12
TPCC.ORDERS_144_DSI O_12
TPCC.ORDERS_145_DSI O_1
TPCC.ORDERS_146_DSI O_1
TPCC.ORDERS_147_DSI O_1
TPCC.ORDERS_148_DSI O_1
TPCC.ORDERS_149_DSI O_1
TPCC.ORDERS_150_DSI O_1
TPCC.ORDERS_151_DSI O_2
TPCC.ORDERS_152_DSI O_2
TPCC.ORDERS_153_DSI O_2
TPCC.ORDERS_154_DSI O_2
TPCC.ORDERS_155_DSI O_2
TPCC.ORDERS_156_DSI O_2
TPCC.ORDERS_157_DSI O_3
TPCC.ORDERS_158_DSI O_3
TPCC.ORDERS_159_DSI O_3
TPCC.ORDERS_160_DSI O_3
TPCC.ORDERS_161_DSI O_3
TPCC.ORDERS_162_DSI O_3
TPCC.ORDERS_163_DSI O_4
TPCC.ORDERS_164_DSI O_4
TPCC.ORDERS_165_DSI O_4
TPCC.ORDERS_166_DSI O_4
TPCC.ORDERS_167_DSI O_4
TPCC.ORDERS_168_DSI O_4
TPCC.ORDERS_169_DSI O_5
TPCC.ORDERS_170_DSI O_5
TPCC.ORDERS_171_DSI O_5
TPCC.ORDERS_172_DSI O_5
TPCC.ORDERS_173_DSI O_5
TPCC.ORDERS_174_DSI O_5
TPCC.ORDERS_175_DSI O_6
TPCC.ORDERS_176_DSI O_6
TPCC.ORDERS_177_DSI O_6
TPCC.ORDERS_178_DSI O_6
TPCC.ORDERS_179_DSI O_6
TPCC.ORDERS_180_DSI O_6
TPCC.ORDERS_181_DSI O_7
TPCC.ORDERS_182_DSI O_7
TPCC.ORDERS_183_DSI O_7
TPCC.ORDERS_184_DSI O_7
TPCC.ORDERS_185_DSI O_7
TPCC.ORDERS_186_DSI O_7
TPCC.ORDERS_187_DSI O_8
TPCC.ORDERS_188_DSI O_8
TPCC.ORDERS_189_DSI O_8
TPCC.ORDERS_190_DSI O_8
TPCC.ORDERS_191_DSI O_8
TPCC.ORDERS_192_DSI O_8
TPCC.ORDERS_193_DSI O_9
TPCC.ORDERS_194_DSI O_9
TPCC.ORDERS_195_DSI O_9
TPCC.ORDERS_196_DSI O_9
TPCC.ORDERS_197_DSI O_9
TPCC.ORDERS_198_DSI O_9
TPCC.ORDERS_199_DSI O_10
TPCC.ORDERS_200_DSI O_10
TPCC.ORDERS_201_DSI O_10
TPCC.ORDERS_202_DSI O_10
TPCC.ORDERS_203_DSI O_10
TPCC.ORDERS_204_DSI O_10
TPCC.ORDERS_205_DSI O_11
TPCC.ORDERS_206_DSI O_11
TPCC.ORDERS_207_DSI O_11
TPCC.ORDERS_208_DSI O_11
TPCC.ORDERS_209_DSI O_11
TPCC.ORDERS_210_DSI O_11
TPCC.ORDERS_211_DSI O_12
TPCC.ORDERS_212_DSI O_12
TPCC.ORDERS_213_DSI O_12

TPCC.ORDERS_214_DSI O_12
TPCC.ORDERS_215_DSI O_12
TPCC.ORDERS_216_DSI O_12
TPCC.ORDERS_217_DSI O_1
TPCC.ORDERS_218_DSI O_1
TPCC.ORDERS_219_DSI O_1
TPCC.ORDERS_220_DSI O_1
TPCC.ORDERS_221_DSI O_1
TPCC.ORDERS_222_DSI O_1
TPCC.ORDERS_223_DSI O_2
TPCC.ORDERS_224_DSI O_2
TPCC.ORDERS_225_DSI O_2
TPCC.ORDERS_226_DSI O_2
TPCC.ORDERS_227_DSI O_2
TPCC.ORDERS_228_DSI O_2
TPCC.ORDERS_229_DSI O_7
TPCC.ORDERS_230_DSI O_7
TPCC.ORDERS_231_DSI O_8
TPCC.ORDERS_232_DSI O_8
TPCC.ORDERS_233_DSI O_9
TPCC.ORDERS_234_DSI O_9
TPCC.ORDERS_235_DSI O_10
TPCC.ORDERS_236_DSI O_10
TPCC.ORDERS_237_DSI O_5
TPCC.ORDERS_238_DSI O_5
TPCC.ORDERS_239_DSI O_6
TPCC.ORDERS_240_DSI O_6
TPCC.ORDERS_241_DSI O_3
TPCC.ORDERS_242_DSI O_3
TPCC.ORDERS_243_DSI O_4
TPCC.ORDERS_244_DSI O_4
TPCC.ORDERS_245_DSI O_5
TPCC.ORDERS_246_DSI O_5
TPCC.ORDERS_247_DSI O_6
TPCC.ORDERS_248_DSI O_6
TPCC.ORDERS_249_DSI O_1
TPCC.ORDERS_250_DSI O_1
TPCC.ORDERS_251_DSI O_2
TPCC.ORDERS_252_DSI O_2
TPCC.ORDERS_IX_1_DSI O_IX_1
TPCC.ORDERS_IX_2_DSI O_IX_1
TPCC.ORDERS_IX_3_DSI O_IX_1
TPCC.ORDERS_IX_4_DSI O_IX_1
TPCC.ORDERS_IX_5_DSI O_IX_1
TPCC.ORDERS_IX_6_DSI O_IX_1
TPCC.ORDERS_IX_7_DSI O_IX_2
TPCC.ORDERS_IX_8_DSI O_IX_2
TPCC.ORDERS_IX_9_DSI O_IX_2
TPCC.ORDERS_IX_10_DSI O_IX_2
TPCC.ORDERS_IX_11_DSI O_IX_2
TPCC.ORDERS_IX_12_DSI O_IX_2
TPCC.ORDERS_IX_13_DSI O_IX_3
TPCC.ORDERS_IX_14_DSI O_IX_3
TPCC.ORDERS_IX_15_DSI O_IX_3
TPCC.ORDERS_IX_16_DSI O_IX_3
TPCC.ORDERS_IX_17_DSI O_IX_3
TPCC.ORDERS_IX_18_DSI O_IX_3
TPCC.ORDERS_IX_19_DSI O_IX_4
TPCC.ORDERS_IX_20_DSI O_IX_4
TPCC.ORDERS_IX_21_DSI O_IX_4
TPCC.ORDERS_IX_22_DSI O_IX_4
TPCC.ORDERS_IX_23_DSI O_IX_4
TPCC.ORDERS_IX_24_DSI O_IX_4
TPCC.ORDERS_IX_25_DSI O_IX_5
TPCC.ORDERS_IX_26_DSI O_IX_5
TPCC.ORDERS_IX_27_DSI O_IX_5
TPCC.ORDERS_IX_28_DSI O_IX_5
TPCC.ORDERS_IX_29_DSI O_IX_5
TPCC.ORDERS_IX_30_DSI O_IX_5
TPCC.ORDERS_IX_31_DSI O_IX_6
TPCC.ORDERS_IX_32_DSI O_IX_6
TPCC.ORDERS_IX_33_DSI O_IX_6
TPCC.ORDERS_IX_34_DSI O_IX_6
TPCC.ORDERS_IX_35_DSI O_IX_6
TPCC.ORDERS_IX_36_DSI O_IX_6
TPCC.ORDERS_IX_37_DSI O_IX_7
TPCC.ORDERS_IX_38_DSI O_IX_7
TPCC.ORDERS_IX_39_DSI O_IX_7
TPCC.ORDERS_IX_40_DSI O_IX_7
TPCC.ORDERS_IX_41_DSI O_IX_7
TPCC.ORDERS_IX_42_DSI O_IX_7
TPCC.ORDERS_IX_43_DSI O_IX_8
TPCC.ORDERS_IX_44_DSI O_IX_8
TPCC.ORDERS_IX_45_DSI O_IX_8
TPCC.ORDERS_IX_46_DSI O_IX_8
TPCC.ORDERS_IX_47_DSI O_IX_8
TPCC.ORDERS_IX_48_DSI O_IX_8
TPCC.ORDERS_IX_49_DSI O_IX_9
TPCC.ORDERS_IX_50_DSI O_IX_9
TPCC.ORDERS_IX_51_DSI O_IX_9
TPCC.ORDERS_IX_52_DSI O_IX_9
TPCC.ORDERS_IX_53_DSI O_IX_9
TPCC.ORDERS_IX_54_DSI O_IX_9
TPCC.ORDERS_IX_55_DSI O_IX_10
TPCC.ORDERS_IX_56_DSI O_IX_10
TPCC.ORDERS_IX_57_DSI O_IX_10
TPCC.ORDERS_IX_58_DSI O_IX_10
TPCC.ORDERS_IX_59_DSI O_IX_10
TPCC.ORDERS_IX_60_DSI O_IX_10
TPCC.ORDERS_IX_61_DSI O_IX_11
TPCC.ORDERS_IX_62_DSI O_IX_11
TPCC.ORDERS_IX_63_DSI O_IX_11
TPCC.ORDERS_IX_64_DSI O_IX_11
TPCC.ORDERS_IX_65_DSI O_IX_11
TPCC.ORDERS_IX_66_DSI O_IX_11
TPCC.ORDERS_IX_67_DSI O_IX_12
TPCC.ORDERS_IX_68_DSI O_IX_12
TPCC.ORDERS_IX_69_DSI O_IX_12
TPCC.ORDERS_IX_70_DSI O_IX_12
TPCC.ORDERS_IX_71_DSI O_IX_12
TPCC.ORDERS_IX_72_DSI O_IX_12
TPCC.ORDERS_IX_73_DSI O_IX_1
TPCC.ORDERS_IX_74_DSI O_IX_1
TPCC.ORDERS_IX_75_DSI O_IX_1
TPCC.ORDERS_IX_76_DSI O_IX_1
TPCC.ORDERS_IX_77_DSI O_IX_1
TPCC.ORDERS_IX_78_DSI O_IX_1
TPCC.ORDERS_IX_79_DSI O_IX_2
TPCC.ORDERS_IX_80_DSI O_IX_2
TPCC.ORDERS_IX_81_DSI O_IX_2
TPCC.ORDERS_IX_82_DSI O_IX_2
TPCC.ORDERS_IX_83_DSI O_IX_2
TPCC.ORDERS_IX_84_DSI O_IX_2
TPCC.ORDERS_IX_85_DSI O_IX_3
TPCC.ORDERS_IX_86_DSI O_IX_3
TPCC.ORDERS_IX_87_DSI O_IX_3
TPCC.ORDERS_IX_88_DSI O_IX_3
TPCC.ORDERS_IX_89_DSI O_IX_3
TPCC.ORDERS_IX_90_DSI O_IX_3
TPCC.ORDERS_IX_91_DSI O_IX_4
TPCC.ORDERS_IX_92_DSI O_IX_4
TPCC.ORDERS_IX_93_DSI O_IX_4
TPCC.ORDERS_IX_94_DSI O_IX_4
TPCC.ORDERS_IX_95_DSI O_IX_4
TPCC.ORDERS_IX_96_DSI O_IX_4
TPCC.ORDERS_IX_97_DSI O_IX_5
TPCC.ORDERS_IX_98_DSI O_IX_5
TPCC.ORDERS_IX_99_DSI O_IX_5
TPCC.ORDERS_IX_100_DSI O_IX_5
TPCC.ORDERS_IX_101_DSI O_IX_5
TPCC.ORDERS_IX_102_DSI O_IX_5
TPCC.ORDERS_IX_103_DSI O_IX_6
TPCC.ORDERS_IX_104_DSI O_IX_6
TPCC.ORDERS_IX_105_DSI O_IX_6
TPCC.ORDERS_IX_106_DSI O_IX_6
TPCC.ORDERS_IX_107_DSI O_IX_6
TPCC.ORDERS_IX_108_DSI O_IX_6
TPCC.ORDERS_IX_109_DSI O_IX_7
TPCC.ORDERS_IX_110_DSI O_IX_7
TPCC.ORDERS_IX_111_DSI O_IX_7
TPCC.ORDERS_IX_112_DSI O_IX_7
TPCC.ORDERS_IX_113_DSI O_IX_7
TPCC.ORDERS_IX_114_DSI O_IX_7
TPCC.ORDERS_IX_115_DSI O_IX_8
TPCC.ORDERS_IX_116_DSI O_IX_8
TPCC.ORDERS_IX_117_DSI O_IX_8
TPCC.ORDERS_IX_118_DSI O_IX_8
TPCC.ORDERS_IX_119_DSI O_IX_8
TPCC.ORDERS_IX_120_DSI O_IX_8
TPCC.ORDERS_IX_121_DSI O_IX_9
TPCC.ORDERS_IX_122_DSI O_IX_9
TPCC.ORDERS_IX_123_DSI O_IX_9
TPCC.ORDERS_IX_124_DSI O_IX_9
TPCC.ORDERS_IX_125_DSI O_IX_9
TPCC.ORDERS_IX_126_DSI O_IX_9
TPCC.ORDERS_IX_127_DSI O_IX_10
TPCC.ORDERS_IX_128_DSI O_IX_10
TPCC.ORDERS_IX_129_DSI O_IX_10
TPCC.ORDERS_IX_130_DSI O_IX_10
TPCC.ORDERS_IX_131_DSI O_IX_10
TPCC.ORDERS_IX_132_DSI O_IX_10
TPCC.ORDERS_IX_133_DSI O_IX_11
TPCC.ORDERS_IX_134_DSI O_IX_11
TPCC.ORDERS_IX_135_DSI O_IX_11
TPCC.ORDERS_IX_136_DSI O_IX_11
TPCC.ORDERS_IX_137_DSI O_IX_11
TPCC.ORDERS_IX_138_DSI O_IX_11
TPCC.ORDERS_IX_139_DSI O_IX_12
TPCC.ORDERS_IX_140_DSI O_IX_12
TPCC.ORDERS_IX_141_DSI O_IX_12
TPCC.ORDERS_IX_142_DSI O_IX_12
TPCC.ORDERS_IX_143_DSI O_IX_12
TPCC.ORDERS_IX_144_DSI O_IX_12
TPCC.ORDERS_IX_145_DSI O_IX_1
TPCC.ORDERS_IX_146_DSI O_IX_1
TPCC.ORDERS_IX_147_DSI O_IX_1
TPCC.ORDERS_IX_148_DSI O_IX_1
TPCC.ORDERS_IX_149_DSI O_IX_1
TPCC.ORDERS_IX_150_DSI O_IX_1
TPCC.ORDERS_IX_151_DSI O_IX_2
TPCC.ORDERS_IX_152_DSI O_IX_2
TPCC.ORDERS_IX_153_DSI O_IX_2
TPCC.ORDERS_IX_154_DSI O_IX_2
TPCC.ORDERS_IX_155_DSI O_IX_2
TPCC.ORDERS_IX_156_DSI O_IX_2
TPCC.ORDERS_IX_157_DSI O_IX_3
TPCC.ORDERS_IX_158_DSI O_IX_3
TPCC.ORDERS_IX_159_DSI O_IX_3
TPCC.ORDERS_IX_160_DSI O_IX_3
TPCC.ORDERS_IX_161_DSI O_IX_3
TPCC.ORDERS_IX_162_DSI O_IX_3
TPCC.ORDERS_IX_163_DSI O_IX_4
TPCC.ORDERS_IX_164_DSI O_IX_4
TPCC.ORDERS_IX_165_DSI O_IX_4

TPCC.ORDERS_IX_166_DSI O_IX_4
TPCC.ORDERS_IX_167_DSI O_IX_4
TPCC.ORDERS_IX_168_DSI O_IX_4
TPCC.ORDERS_IX_169_DSI O_IX_5
TPCC.ORDERS_IX_170_DSI O_IX_5
TPCC.ORDERS_IX_171_DSI O_IX_5
TPCC.ORDERS_IX_172_DSI O_IX_5
TPCC.ORDERS_IX_173_DSI O_IX_5
TPCC.ORDERS_IX_174_DSI O_IX_5
TPCC.ORDERS_IX_175_DSI O_IX_6
TPCC.ORDERS_IX_176_DSI O_IX_6
TPCC.ORDERS_IX_177_DSI O_IX_6
TPCC.ORDERS_IX_178_DSI O_IX_6
TPCC.ORDERS_IX_179_DSI O_IX_6
TPCC.ORDERS_IX_180_DSI O_IX_6
TPCC.ORDERS_IX_181_DSI O_IX_7
TPCC.ORDERS_IX_182_DSI O_IX_7
TPCC.ORDERS_IX_183_DSI O_IX_7
TPCC.ORDERS_IX_184_DSI O_IX_7
TPCC.ORDERS_IX_185_DSI O_IX_7
TPCC.ORDERS_IX_186_DSI O_IX_7
TPCC.ORDERS_IX_187_DSI O_IX_8
TPCC.ORDERS_IX_188_DSI O_IX_8
TPCC.ORDERS_IX_189_DSI O_IX_8
TPCC.ORDERS_IX_190_DSI O_IX_8
TPCC.ORDERS_IX_191_DSI O_IX_8
TPCC.ORDERS_IX_192_DSI O_IX_8
TPCC.ORDERS_IX_193_DSI O_IX_9
TPCC.ORDERS_IX_194_DSI O_IX_9
TPCC.ORDERS_IX_195_DSI O_IX_9
TPCC.ORDERS_IX_196_DSI O_IX_9
TPCC.ORDERS_IX_197_DSI O_IX_9
TPCC.ORDERS_IX_198_DSI O_IX_9
TPCC.ORDERS_IX_199_DSI O_IX_10
TPCC.ORDERS_IX_200_DSI O_IX_10
TPCC.ORDERS_IX_201_DSI O_IX_10
TPCC.ORDERS_IX_202_DSI O_IX_10
TPCC.ORDERS_IX_203_DSI O_IX_10
TPCC.ORDERS_IX_204_DSI O_IX_10
TPCC.ORDERS_IX_205_DSI O_IX_11
TPCC.ORDERS_IX_206_DSI O_IX_11
TPCC.ORDERS_IX_207_DSI O_IX_11
TPCC.ORDERS_IX_208_DSI O_IX_11
TPCC.ORDERS_IX_209_DSI O_IX_11
TPCC.ORDERS_IX_210_DSI O_IX_11
TPCC.ORDERS_IX_211_DSI O_IX_12
TPCC.ORDERS_IX_212_DSI O_IX_12
TPCC.ORDERS_IX_213_DSI O_IX_12
TPCC.ORDERS_IX_214_DSI O_IX_12
TPCC.ORDERS_IX_215_DSI O_IX_12
TPCC.ORDERS_IX_216_DSI O_IX_12
TPCC.ORDERS_IX_217_DSI O_IX_1
TPCC.ORDERS_IX_218_DSI O_IX_1
TPCC.ORDERS_IX_219_DSI O_IX_1
TPCC.ORDERS_IX_220_DSI O_IX_1
TPCC.ORDERS_IX_221_DSI O_IX_1
TPCC.ORDERS_IX_222_DSI O_IX_1
TPCC.ORDERS_IX_223_DSI O_IX_2
TPCC.ORDERS_IX_224_DSI O_IX_2
TPCC.ORDERS_IX_225_DSI O_IX_2
TPCC.ORDERS_IX_226_DSI O_IX_2
TPCC.ORDERS_IX_227_DSI O_IX_2
TPCC.ORDERS_IX_228_DSI O_IX_2
TPCC.ORDERS_IX_229_DSI O_IX_7
TPCC.ORDERS_IX_230_DSI O_IX_7
TPCC.ORDERS_IX_231_DSI O_IX_8
TPCC.ORDERS_IX_232_DSI O_IX_8
TPCC.ORDERS_IX_233_DSI O_IX_9
TPCC.ORDERS_IX_234_DSI O_IX_9
TPCC.ORDERS_IX_235_DSI O_IX_10
TPCC.ORDERS_IX_236_DSI O_IX_10
TPCC.ORDERS_IX_237_DSI O_IX_5
TPCC.ORDERS_IX_238_DSI O_IX_5
TPCC.ORDERS_IX_239_DSI O_IX_6
TPCC.ORDERS_IX_240_DSI O_IX_6
TPCC.ORDERS_IX_241_DSI O_IX_3
TPCC.ORDERS_IX_242_DSI O_IX_3
TPCC.ORDERS_IX_243_DSI O_IX_4
TPCC.ORDERS_IX_244_DSI O_IX_4
TPCC.ORDERS_IX_245_DSI O_IX_5
TPCC.ORDERS_IX_246_DSI O_IX_5
TPCC.ORDERS_IX_247_DSI O_IX_6
TPCC.ORDERS_IX_248_DSI O_IX_6
TPCC.ORDERS_IX_249_DSI O_IX_1
TPCC.ORDERS_IX_250_DSI O_IX_1
TPCC.ORDERS_IX_251_DSI O_IX_2
TPCC.ORDERS_IX_252_DSI O_IX_2
TPCC.NEWORDER_1_DSI NO_1
TPCC.NEWORDER_2_DSI NO_2
TPCC.NEWORDER_3_DSI NO_3
TPCC.NEWORDER_4_DSI NO_4
TPCC.NEWORDER_5_DSI NO_5
TPCC.NEWORDER_6_DSI NO_6
TPCC.NEWORDER_7_DSI NO_7
TPCC.NEWORDER_8_DSI NO_8
TPCC.NEWORDER_9_DSI NO_9
TPCC.NEWORDER_10_DSI NO_10
TPCC.NEWORDER_11_DSI NO_11
TPCC.NEWORDER_12_DSI NO_12
TPCC.NEWORDER_13_DSI NO_1
TPCC.NEWORDER_14_DSI NO_2
TPCC.NEWORDER_15_DSI NO_3
TPCC.NEWORDER_16_DSI NO_4
TPCC.NEWORDER_17_DSI NO_5
TPCC.NEWORDER_18_DSI NO_6
TPCC.NEWORDER_19_DSI NO_7
TPCC.NEWORDER_20_DSI NO_8
TPCC.NEWORDER_21_DSI NO_9
TPCC.NEWORDER_22_DSI NO_10
TPCC.NEWORDER_23_DSI NO_11
TPCC.NEWORDER_24_DSI NO_12
TPCC.NEWORDER_25_DSI NO_1
TPCC.NEWORDER_26_DSI NO_2
TPCC.NEWORDER_27_DSI NO_3
TPCC.NEWORDER_28_DSI NO_4
TPCC.NEWORDER_29_DSI NO_5
TPCC.NEWORDER_30_DSI NO_6
TPCC.NEWORDER_31_DSI NO_7
TPCC.NEWORDER_32_DSI NO_8
TPCC.NEWORDER_33_DSI NO_9
TPCC.NEWORDER_34_DSI NO_10
TPCC.NEWORDER_35_DSI NO_11
TPCC.NEWORDER_36_DSI NO_12
TPCC.NEWORDER_37_DSI NO_1
TPCC.NEWORDER_38_DSI NO_2
TPCC.NEWORDER_39_DSI NO_3
TPCC.NEWORDER_40_DSI NO_4
TPCC.NEWORDER_41_DSI NO_5
TPCC.NEWORDER_42_DSI NO_6
TPCC.NEWORDER_43_DSI NO_7
TPCC.NEWORDER_44_DSI NO_8
TPCC.NEWORDER_45_DSI NO_9
TPCC.NEWORDER_46_DSI NO_10
TPCC.NEWORDER_47_DSI NO_11
TPCC.NEWORDER_48_DSI NO_12
TPCC.NEWORDER_49_DSI NO_1
TPCC.NEWORDER_50_DSI NO_2
TPCC.NEWORDER_51_DSI NO_3
TPCC.NEWORDER_52_DSI NO_4
TPCC.NEWORDER_53_DSI NO_5
TPCC.NEWORDER_54_DSI NO_6
TPCC.NEWORDER_55_DSI NO_7
TPCC.NEWORDER_56_DSI NO_8
TPCC.NEWORDER_57_DSI NO_9
TPCC.NEWORDER_58_DSI NO_10
TPCC.NEWORDER_59_DSI NO_11
TPCC.NEWORDER_60_DSI NO_12
TPCC.NEWORDER_61_DSI NO_1
TPCC.NEWORDER_62_DSI NO_2
TPCC.NEWORDER_63_DSI NO_3
TPCC.NEWORDER_64_DSI NO_4
TPCC.NEWORDER_65_DSI NO_5
TPCC.NEWORDER_66_DSI NO_6
TPCC.NEWORDER_67_DSI NO_7
TPCC.NEWORDER_68_DSI NO_8
TPCC.NEWORDER_69_DSI NO_9
TPCC.NEWORDER_70_DSI NO_10
TPCC.NEWORDER_71_DSI NO_11
TPCC.NEWORDER_72_DSI NO_12
TPCC.NEWORDER_73_DSI NO_1
TPCC.NEWORDER_74_DSI NO_2
TPCC.NEWORDER_75_DSI NO_3
TPCC.NEWORDER_76_DSI NO_4
TPCC.NEWORDER_77_DSI NO_5
TPCC.NEWORDER_78_DSI NO_6
TPCC.NEWORDER_79_DSI NO_7
TPCC.NEWORDER_80_DSI NO_8
TPCC.NEWORDER_81_DSI NO_9
TPCC.NEWORDER_82_DSI NO_10
TPCC.NEWORDER_83_DSI NO_11
TPCC.NEWORDER_84_DSI NO_12
TPCC.NEWORDER_85_DSI NO_1
TPCC.NEWORDER_86_DSI NO_2
TPCC.NEWORDER_87_DSI NO_3
TPCC.NEWORDER_88_DSI NO_4
TPCC.NEWORDER_89_DSI NO_5
TPCC.NEWORDER_90_DSI NO_6
TPCC.NEWORDER_91_DSI NO_7
TPCC.NEWORDER_92_DSI NO_8
TPCC.NEWORDER_93_DSI NO_9
TPCC.NEWORDER_94_DSI NO_10
TPCC.NEWORDER_95_DSI NO_11
TPCC.NEWORDER_96_DSI NO_12
TPCC.NEWORDER_97_DSI NO_1
TPCC.NEWORDER_98_DSI NO_2
TPCC.NEWORDER_99_DSI NO_3
TPCC.NEWORDER_100_DSI NO_4
TPCC.NEWORDER_101_DSI NO_5
TPCC.NEWORDER_102_DSI NO_6
TPCC.NEWORDER_103_DSI NO_7
TPCC.NEWORDER_104_DSI NO_8
TPCC.NEWORDER_105_DSI NO_9
TPCC.NEWORDER_106_DSI NO_10
TPCC.NEWORDER_107_DSI NO_11
TPCC.NEWORDER_108_DSI NO_12
TPCC.NEWORDER_109_DSI NO_1
TPCC.NEWORDER_110_DSI NO_2
TPCC.NEWORDER_111_DSI NO_3
TPCC.NEWORDER_112_DSI NO_4
TPCC.NEWORDER_113_DSI NO_5
TPCC.NEWORDER_114_DSI NO_6
TPCC.NEWORDER_115_DSI NO_7
TPCC.NEWORDER_116_DSI NO_8
TPCC.NEWORDER_117_DSI NO_9

TPCC.NEWORDER_118_DSI NO_10
TPCC.NEWORDER_119_DSI NO_11
TPCC.NEWORDER_120_DSI NO_12
TPCC.NEWORDER_121_DSI NO_1
TPCC.NEWORDER_122_DSI NO_2
TPCC.NEWORDER_123_DSI NO_3
TPCC.NEWORDER_124_DSI NO_4
TPCC.NEWORDER_125_DSI NO_5
TPCC.NEWORDER_126_DSI NO_6
TPCC.NEWORDER_127_DSI NO_7
TPCC.NEWORDER_128_DSI NO_8
TPCC.NEWORDER_129_DSI NO_9
TPCC.NEWORDER_130_DSI NO_10
TPCC.NEWORDER_131_DSI NO_11
TPCC.NEWORDER_132_DSI NO_12
TPCC.NEWORDER_133_DSI NO_1
TPCC.NEWORDER_134_DSI NO_2
TPCC.NEWORDER_135_DSI NO_3
TPCC.NEWORDER_136_DSI NO_4
TPCC.NEWORDER_137_DSI NO_5
TPCC.NEWORDER_138_DSI NO_6
TPCC.NEWORDER_139_DSI NO_7
TPCC.NEWORDER_140_DSI NO_8
TPCC.NEWORDER_141_DSI NO_9
TPCC.NEWORDER_142_DSI NO_10
TPCC.NEWORDER_143_DSI NO_11
TPCC.NEWORDER_144_DSI NO_12
TPCC.NEWORDER_145_DSI NO_1
TPCC.NEWORDER_146_DSI NO_2
TPCC.NEWORDER_147_DSI NO_3
TPCC.NEWORDER_148_DSI NO_4
TPCC.NEWORDER_149_DSI NO_5
TPCC.NEWORDER_150_DSI NO_6
TPCC.NEWORDER_151_DSI NO_7
TPCC.NEWORDER_152_DSI NO_8
TPCC.NEWORDER_153_DSI NO_9
TPCC.NEWORDER_154_DSI NO_10
TPCC.NEWORDER_155_DSI NO_11
TPCC.NEWORDER_156_DSI NO_12
TPCC.NEWORDER_157_DSI NO_1
TPCC.NEWORDER_158_DSI NO_2
TPCC.NEWORDER_159_DSI NO_3
TPCC.NEWORDER_160_DSI NO_4
TPCC.NEWORDER_161_DSI NO_5
TPCC.NEWORDER_162_DSI NO_6
TPCC.NEWORDER_163_DSI NO_7
TPCC.NEWORDER_164_DSI NO_8
TPCC.NEWORDER_165_DSI NO_9
TPCC.NEWORDER_166_DSI NO_10
TPCC.NEWORDER_167_DSI NO_11
TPCC.NEWORDER_168_DSI NO_12
TPCC.NEWORDER_169_DSI NO_1
TPCC.NEWORDER_170_DSI NO_2
TPCC.NEWORDER_171_DSI NO_3
TPCC.NEWORDER_172_DSI NO_4
TPCC.NEWORDER_173_DSI NO_5
TPCC.NEWORDER_174_DSI NO_6
TPCC.NEWORDER_175_DSI NO_7
TPCC.NEWORDER_176_DSI NO_8
TPCC.NEWORDER_177_DSI NO_9
TPCC.NEWORDER_178_DSI NO_10
TPCC.NEWORDER_179_DSI NO_11
TPCC.NEWORDER_180_DSI NO_12
TPCC.NEWORDER_181_DSI NO_1
TPCC.NEWORDER_182_DSI NO_2
TPCC.NEWORDER_183_DSI NO_3
TPCC.NEWORDER_184_DSI NO_4
TPCC.NEWORDER_185_DSI NO_5
TPCC.NEWORDER_186_DSI NO_6
TPCC.NEWORDER_187_DSI NO_7
TPCC.NEWORDER_188_DSI NO_8
TPCC.NEWORDER_189_DSI NO_9
TPCC.NEWORDER_190_DSI NO_10
TPCC.NEWORDER_191_DSI NO_11
TPCC.NEWORDER_192_DSI NO_12
TPCC.NEWORDER_193_DSI NO_1
TPCC.NEWORDER_194_DSI NO_2
TPCC.NEWORDER_195_DSI NO_3
TPCC.NEWORDER_196_DSI NO_4
TPCC.NEWORDER_197_DSI NO_5
TPCC.NEWORDER_198_DSI NO_6
TPCC.NEWORDER_199_DSI NO_7
TPCC.NEWORDER_200_DSI NO_8
TPCC.NEWORDER_201_DSI NO_9
TPCC.NEWORDER_202_DSI NO_10
TPCC.NEWORDER_203_DSI NO_11
TPCC.NEWORDER_204_DSI NO_12
TPCC.NEWORDER_205_DSI NO_1
TPCC.NEWORDER_206_DSI NO_2
TPCC.NEWORDER_207_DSI NO_3
TPCC.NEWORDER_208_DSI NO_4
TPCC.NEWORDER_209_DSI NO_5
TPCC.NEWORDER_210_DSI NO_6
TPCC.NEWORDER_211_DSI NO_7
TPCC.NEWORDER_212_DSI NO_8
TPCC.NEWORDER_213_DSI NO_9
TPCC.NEWORDER_214_DSI NO_10
TPCC.NEWORDER_215_DSI NO_11
TPCC.NEWORDER_216_DSI NO_12
TPCC.NEWORDER_217_DSI NO_1
TPCC.NEWORDER_218_DSI NO_2
TPCC.NEWORDER_219_DSI NO_3
TPCC.NEWORDER_220_DSI NO_4
TPCC.NEWORDER_221_DSI NO_5
TPCC.NEWORDER_222_DSI NO_6
TPCC.NEWORDER_223_DSI NO_7
TPCC.NEWORDER_224_DSI NO_8
TPCC.NEWORDER_225_DSI NO_9
TPCC.NEWORDER_226_DSI NO_10
TPCC.NEWORDER_227_DSI NO_11
TPCC.NEWORDER_228_DSI NO_12
TPCC.NEWORDER_229_DSI NO_1
TPCC.NEWORDER_230_DSI NO_2
TPCC.NEWORDER_231_DSI NO_3
TPCC.NEWORDER_232_DSI NO_4
TPCC.NEWORDER_233_DSI NO_5
TPCC.NEWORDER_234_DSI NO_6
TPCC.NEWORDER_235_DSI NO_7
TPCC.NEWORDER_236_DSI NO_8
TPCC.NEWORDER_237_DSI NO_9
TPCC.NEWORDER_238_DSI NO_10
TPCC.NEWORDER_239_DSI NO_11
TPCC.NEWORDER_240_DSI NO_12
TPCC.NEWORDER_241_DSI NO_1
TPCC.NEWORDER_242_DSI NO_2
TPCC.NEWORDER_243_DSI NO_3
TPCC.NEWORDER_244_DSI NO_4
TPCC.NEWORDER_245_DSI NO_5
TPCC.NEWORDER_246_DSI NO_6
TPCC.NEWORDER_247_DSI NO_7
TPCC.NEWORDER_248_DSI NO_8
TPCC.NEWORDER_249_DSI NO_9
TPCC.NEWORDER_250_DSI NO_10
TPCC.NEWORDER_251_DSI NO_11
TPCC.NEWORDER_252_DSI NO_12
TPCC.NEWORDER_X_1_DSI NO_IX_1
TPCC.NEWORDER_X_2_DSI NO_IX_2
TPCC.NEWORDER_X_3_DSI NO_IX_3
TPCC.NEWORDER_X_4_DSI NO_IX_4
TPCC.NEWORDER_X_5_DSI NO_IX_5
TPCC.NEWORDER_X_6_DSI NO_IX_6
TPCC.NEWORDER_X_7_DSI NO_IX_7
TPCC.NEWORDER_X_8_DSI NO_IX_8
TPCC.NEWORDER_X_9_DSI NO_IX_9
TPCC.NEWORDER_X_10_DSI NO_IX_10
TPCC.NEWORDER_X_11_DSI NO_IX_11
TPCC.NEWORDER_X_12_DSI NO_IX_12
TPCC.NEWORDER_X_13_DSI NO_IX_1
TPCC.NEWORDER_X_14_DSI NO_IX_2
TPCC.NEWORDER_X_15_DSI NO_IX_3
TPCC.NEWORDER_X_16_DSI NO_IX_4
TPCC.NEWORDER_X_17_DSI NO_IX_5
TPCC.NEWORDER_X_18_DSI NO_IX_6
TPCC.NEWORDER_X_19_DSI NO_IX_7
TPCC.NEWORDER_X_20_DSI NO_IX_8
TPCC.NEWORDER_X_21_DSI NO_IX_9
TPCC.NEWORDER_X_22_DSI NO_IX_10
TPCC.NEWORDER_X_23_DSI NO_IX_11
TPCC.NEWORDER_X_24_DSI NO_IX_12
TPCC.NEWORDER_X_25_DSI NO_IX_1
TPCC.NEWORDER_X_26_DSI NO_IX_2
TPCC.NEWORDER_X_27_DSI NO_IX_3
TPCC.NEWORDER_X_28_DSI NO_IX_4
TPCC.NEWORDER_X_29_DSI NO_IX_5
TPCC.NEWORDER_X_30_DSI NO_IX_6
TPCC.NEWORDER_X_31_DSI NO_IX_7
TPCC.NEWORDER_X_32_DSI NO_IX_8
TPCC.NEWORDER_X_33_DSI NO_IX_9
TPCC.NEWORDER_X_34_DSI NO_IX_10
TPCC.NEWORDER_X_35_DSI NO_IX_11
TPCC.NEWORDER_X_36_DSI NO_IX_12
TPCC.NEWORDER_X_37_DSI NO_IX_1
TPCC.NEWORDER_X_38_DSI NO_IX_2
TPCC.NEWORDER_X_39_DSI NO_IX_3
TPCC.NEWORDER_X_40_DSI NO_IX_4
TPCC.NEWORDER_X_41_DSI NO_IX_5
TPCC.NEWORDER_X_42_DSI NO_IX_6
TPCC.NEWORDER_X_43_DSI NO_IX_7
TPCC.NEWORDER_X_44_DSI NO_IX_8
TPCC.NEWORDER_X_45_DSI NO_IX_9
TPCC.NEWORDER_X_46_DSI NO_IX_10
TPCC.NEWORDER_X_47_DSI NO_IX_11
TPCC.NEWORDER_X_48_DSI NO_IX_12
TPCC.NEWORDER_X_49_DSI NO_IX_1
TPCC.NEWORDER_X_50_DSI NO_IX_2
TPCC.NEWORDER_X_51_DSI NO_IX_3
TPCC.NEWORDER_X_52_DSI NO_IX_4
TPCC.NEWORDER_X_53_DSI NO_IX_5
TPCC.NEWORDER_X_54_DSI NO_IX_6
TPCC.NEWORDER_X_55_DSI NO_IX_7
TPCC.NEWORDER_X_56_DSI NO_IX_8
TPCC.NEWORDER_X_57_DSI NO_IX_9
TPCC.NEWORDER_X_58_DSI NO_IX_10
TPCC.NEWORDER_X_59_DSI NO_IX_11
TPCC.NEWORDER_X_60_DSI NO_IX_12
TPCC.NEWORDER_X_61_DSI NO_IX_1
TPCC.NEWORDER_X_62_DSI NO_IX_2
TPCC.NEWORDER_X_63_DSI NO_IX_3
TPCC.NEWORDER_X_64_DSI NO_IX_4
TPCC.NEWORDER_X_65_DSI NO_IX_5
TPCC.NEWORDER_X_66_DSI NO_IX_6
TPCC.NEWORDER_X_67_DSI NO_IX_7
TPCC.NEWORDER_X_68_DSI NO_IX_8
TPCC.NEWORDER_X_69_DSI NO_IX_9

TPCC.NEWORDER_X_70_DSI NO_IX_10
TPCC.NEWORDER_X_71_DSI NO_IX_11
TPCC.NEWORDER_X_72_DSI NO_IX_12
TPCC.NEWORDER_X_73_DSI NO_IX_1
TPCC.NEWORDER_X_74_DSI NO_IX_2
TPCC.NEWORDER_X_75_DSI NO_IX_3
TPCC.NEWORDER_X_76_DSI NO_IX_4
TPCC.NEWORDER_X_77_DSI NO_IX_5
TPCC.NEWORDER_X_78_DSI NO_IX_6
TPCC.NEWORDER_X_79_DSI NO_IX_7
TPCC.NEWORDER_X_80_DSI NO_IX_8
TPCC.NEWORDER_X_81_DSI NO_IX_9
TPCC.NEWORDER_X_82_DSI NO_IX_10
TPCC.NEWORDER_X_83_DSI NO_IX_11
TPCC.NEWORDER_X_84_DSI NO_IX_12
TPCC.NEWORDER_X_85_DSI NO_IX_1
TPCC.NEWORDER_X_86_DSI NO_IX_2
TPCC.NEWORDER_X_87_DSI NO_IX_3
TPCC.NEWORDER_X_88_DSI NO_IX_4
TPCC.NEWORDER_X_89_DSI NO_IX_5
TPCC.NEWORDER_X_90_DSI NO_IX_6
TPCC.NEWORDER_X_91_DSI NO_IX_7
TPCC.NEWORDER_X_92_DSI NO_IX_8
TPCC.NEWORDER_X_93_DSI NO_IX_9
TPCC.NEWORDER_X_94_DSI NO_IX_10
TPCC.NEWORDER_X_95_DSI NO_IX_11
TPCC.NEWORDER_X_96_DSI NO_IX_12
TPCC.NEWORDER_X_97_DSI NO_IX_1
TPCC.NEWORDER_X_98_DSI NO_IX_2
TPCC.NEWORDER_X_99_DSI NO_IX_3
TPCC.NEWORDER_X_100_DSI NO_IX_4
TPCC.NEWORDER_X_101_DSI NO_IX_5
TPCC.NEWORDER_X_102_DSI NO_IX_6
TPCC.NEWORDER_X_103_DSI NO_IX_7
TPCC.NEWORDER_X_104_DSI NO_IX_8
TPCC.NEWORDER_X_105_DSI NO_IX_9
TPCC.NEWORDER_X_106_DSI NO_IX_10
TPCC.NEWORDER_X_107_DSI NO_IX_11
TPCC.NEWORDER_X_108_DSI NO_IX_12
TPCC.NEWORDER_X_109_DSI NO_IX_1
TPCC.NEWORDER_X_110_DSI NO_IX_2
TPCC.NEWORDER_X_111_DSI NO_IX_3
TPCC.NEWORDER_X_112_DSI NO_IX_4
TPCC.NEWORDER_X_113_DSI NO_IX_5
TPCC.NEWORDER_X_114_DSI NO_IX_6
TPCC.NEWORDER_X_115_DSI NO_IX_7
TPCC.NEWORDER_X_116_DSI NO_IX_8
TPCC.NEWORDER_X_117_DSI NO_IX_9
TPCC.NEWORDER_X_118_DSI NO_IX_10
TPCC.NEWORDER_X_119_DSI NO_IX_11
TPCC.NEWORDER_X_120_DSI NO_IX_12
TPCC.NEWORDER_X_121_DSI NO_IX_1
TPCC.NEWORDER_X_122_DSI NO_IX_2
TPCC.NEWORDER_X_123_DSI NO_IX_3
TPCC.NEWORDER_X_124_DSI NO_IX_4
TPCC.NEWORDER_X_125_DSI NO_IX_5
TPCC.NEWORDER_X_126_DSI NO_IX_6
TPCC.NEWORDER_X_127_DSI NO_IX_7
TPCC.NEWORDER_X_128_DSI NO_IX_8
TPCC.NEWORDER_X_129_DSI NO_IX_9
TPCC.NEWORDER_X_130_DSI NO_IX_10
TPCC.NEWORDER_X_131_DSI NO_IX_11
TPCC.NEWORDER_X_132_DSI NO_IX_12
TPCC.NEWORDER_X_133_DSI NO_IX_1
TPCC.NEWORDER_X_134_DSI NO_IX_2
TPCC.NEWORDER_X_135_DSI NO_IX_3
TPCC.NEWORDER_X_136_DSI NO_IX_4
TPCC.NEWORDER_X_137_DSI NO_IX_5
TPCC.NEWORDER_X_138_DSI NO_IX_6
TPCC.NEWORDER_X_139_DSI NO_IX_7
TPCC.NEWORDER_X_140_DSI NO_IX_8
TPCC.NEWORDER_X_141_DSI NO_IX_9
TPCC.NEWORDER_X_142_DSI NO_IX_10
TPCC.NEWORDER_X_143_DSI NO_IX_11
TPCC.NEWORDER_X_144_DSI NO_IX_12
TPCC.NEWORDER_X_145_DSI NO_IX_1
TPCC.NEWORDER_X_146_DSI NO_IX_2
TPCC.NEWORDER_X_147_DSI NO_IX_3
TPCC.NEWORDER_X_148_DSI NO_IX_4
TPCC.NEWORDER_X_149_DSI NO_IX_5
TPCC.NEWORDER_X_150_DSI NO_IX_6
TPCC.NEWORDER_X_151_DSI NO_IX_7
TPCC.NEWORDER_X_152_DSI NO_IX_8
TPCC.NEWORDER_X_153_DSI NO_IX_9
TPCC.NEWORDER_X_154_DSI NO_IX_10
TPCC.NEWORDER_X_155_DSI NO_IX_11
TPCC.NEWORDER_X_156_DSI NO_IX_12
TPCC.NEWORDER_X_157_DSI NO_IX_1
TPCC.NEWORDER_X_158_DSI NO_IX_2
TPCC.NEWORDER_X_159_DSI NO_IX_3
TPCC.NEWORDER_X_160_DSI NO_IX_4
TPCC.NEWORDER_X_161_DSI NO_IX_5
TPCC.NEWORDER_X_162_DSI NO_IX_6
TPCC.NEWORDER_X_163_DSI NO_IX_7
TPCC.NEWORDER_X_164_DSI NO_IX_8
TPCC.NEWORDER_X_165_DSI NO_IX_9
TPCC.NEWORDER_X_166_DSI NO_IX_10
TPCC.NEWORDER_X_167_DSI NO_IX_11
TPCC.NEWORDER_X_168_DSI NO_IX_12
TPCC.NEWORDER_X_169_DSI NO_IX_1
TPCC.NEWORDER_X_170_DSI NO_IX_2
TPCC.NEWORDER_X_171_DSI NO_IX_3
TPCC.NEWORDER_X_172_DSI NO_IX_4
TPCC.NEWORDER_X_173_DSI NO_IX_5
TPCC.NEWORDER_X_174_DSI NO_IX_6
TPCC.NEWORDER_X_175_DSI NO_IX_7
TPCC.NEWORDER_X_176_DSI NO_IX_8
TPCC.NEWORDER_X_177_DSI NO_IX_9
TPCC.NEWORDER_X_178_DSI NO_IX_10
TPCC.NEWORDER_X_179_DSI NO_IX_11
TPCC.NEWORDER_X_180_DSI NO_IX_12
TPCC.NEWORDER_X_181_DSI NO_IX_1
TPCC.NEWORDER_X_182_DSI NO_IX_2
TPCC.NEWORDER_X_183_DSI NO_IX_3
TPCC.NEWORDER_X_184_DSI NO_IX_4
TPCC.NEWORDER_X_185_DSI NO_IX_5
TPCC.NEWORDER_X_186_DSI NO_IX_6
TPCC.NEWORDER_X_187_DSI NO_IX_7
TPCC.NEWORDER_X_188_DSI NO_IX_8
TPCC.NEWORDER_X_189_DSI NO_IX_9
TPCC.NEWORDER_X_190_DSI NO_IX_10
TPCC.NEWORDER_X_191_DSI NO_IX_11
TPCC.NEWORDER_X_192_DSI NO_IX_12
TPCC.NEWORDER_X_193_DSI NO_IX_1
TPCC.NEWORDER_X_194_DSI NO_IX_2
TPCC.NEWORDER_X_195_DSI NO_IX_3
TPCC.NEWORDER_X_196_DSI NO_IX_4
TPCC.NEWORDER_X_197_DSI NO_IX_5
TPCC.NEWORDER_X_198_DSI NO_IX_6
TPCC.NEWORDER_X_199_DSI NO_IX_7
TPCC.NEWORDER_X_200_DSI NO_IX_8
TPCC.NEWORDER_X_201_DSI NO_IX_9
TPCC.NEWORDER_X_202_DSI NO_IX_10
TPCC.NEWORDER_X_203_DSI NO_IX_11
TPCC.NEWORDER_X_204_DSI NO_IX_12
TPCC.NEWORDER_X_205_DSI NO_IX_1
TPCC.NEWORDER_X_206_DSI NO_IX_2
TPCC.NEWORDER_X_207_DSI NO_IX_3
TPCC.NEWORDER_X_208_DSI NO_IX_4
TPCC.NEWORDER_X_209_DSI NO_IX_5
TPCC.NEWORDER_X_210_DSI NO_IX_6
TPCC.NEWORDER_X_211_DSI NO_IX_7
TPCC.NEWORDER_X_212_DSI NO_IX_8
TPCC.NEWORDER_X_213_DSI NO_IX_9
TPCC.NEWORDER_X_214_DSI NO_IX_10
TPCC.NEWORDER_X_215_DSI NO_IX_11
TPCC.NEWORDER_X_216_DSI NO_IX_12
TPCC.NEWORDER_X_217_DSI NO_IX_1
TPCC.NEWORDER_X_218_DSI NO_IX_2
TPCC.NEWORDER_X_219_DSI NO_IX_3
TPCC.NEWORDER_X_220_DSI NO_IX_4
TPCC.NEWORDER_X_221_DSI NO_IX_5
TPCC.NEWORDER_X_222_DSI NO_IX_6
TPCC.NEWORDER_X_223_DSI NO_IX_7
TPCC.NEWORDER_X_224_DSI NO_IX_8
TPCC.NEWORDER_X_225_DSI NO_IX_9
TPCC.NEWORDER_X_226_DSI NO_IX_10
TPCC.NEWORDER_X_227_DSI NO_IX_11
TPCC.NEWORDER_X_228_DSI NO_IX_12
TPCC.NEWORDER_X_229_DSI NO_IX_1
TPCC.NEWORDER_X_230_DSI NO_IX_2
TPCC.NEWORDER_X_231_DSI NO_IX_3
TPCC.NEWORDER_X_232_DSI NO_IX_4
TPCC.NEWORDER_X_233_DSI NO_IX_5
TPCC.NEWORDER_X_234_DSI NO_IX_6
TPCC.NEWORDER_X_235_DSI NO_IX_7
TPCC.NEWORDER_X_236_DSI NO_IX_8
TPCC.NEWORDER_X_237_DSI NO_IX_9
TPCC.NEWORDER_X_238_DSI NO_IX_10
TPCC.NEWORDER_X_239_DSI NO_IX_11
TPCC.NEWORDER_X_240_DSI NO_IX_12
TPCC.NEWORDER_X_241_DSI NO_IX_1
TPCC.NEWORDER_X_242_DSI NO_IX_2
TPCC.NEWORDER_X_243_DSI NO_IX_3
TPCC.NEWORDER_X_244_DSI NO_IX_4
TPCC.NEWORDER_X_245_DSI NO_IX_5
TPCC.NEWORDER_X_246_DSI NO_IX_6
TPCC.NEWORDER_X_247_DSI NO_IX_7
TPCC.NEWORDER_X_248_DSI NO_IX_8
TPCC.NEWORDER_X_249_DSI NO_IX_9
TPCC.NEWORDER_X_250_DSI NO_IX_10
TPCC.NEWORDER_X_251_DSI NO_IX_11
TPCC.NEWORDER_X_252_DSI NO_IX_12
TPCC.STOCK_1_DSI S_1
TPCC.STOCK_2_DSI S_1
TPCC.STOCK_3_DSI S_2
TPCC.STOCK_4_DSI S_2
TPCC.STOCK_5_DSI S_3
TPCC.STOCK_6_DSI S_3
TPCC.STOCK_7_DSI S_4
TPCC.STOCK_8_DSI S_4
TPCC.STOCK_9_DSI S_5
TPCC.STOCK_10_DSI S_5
TPCC.STOCK_11_DSI S_6
TPCC.STOCK_12_DSI S_6
TPCC.STOCK_13_DSI S_7
TPCC.STOCK_14_DSI S_7
TPCC.STOCK_15_DSI S_8
TPCC.STOCK_16_DSI S_8
TPCC.STOCK_17_DSI S_9
TPCC.STOCK_18_DSI S_9
TPCC.STOCK_19_DSI S_10
TPCC.STOCK_20_DSI S_10
TPCC.STOCK_21_DSI S_11

TPCC.STOCK_22_DSI S_11
TPCC.STOCK_23_DSI S_12
TPCC.STOCK_24_DSI S_12
TPCC.STOCK_25_DSI S_1
TPCC.STOCK_26_DSI S_1
TPCC.STOCK_27_DSI S_2
TPCC.STOCK_28_DSI S_2
TPCC.STOCK_29_DSI S_3
TPCC.STOCK_30_DSI S_3
TPCC.STOCK_31_DSI S_4
TPCC.STOCK_32_DSI S_4
TPCC.STOCK_33_DSI S_5
TPCC.STOCK_34_DSI S_5
TPCC.STOCK_35_DSI S_6
TPCC.STOCK_36_DSI S_6
TPCC.STOCK_37_DSI S_7
TPCC.STOCK_38_DSI S_7
TPCC.STOCK_39_DSI S_8
TPCC.STOCK_40_DSI S_8
TPCC.STOCK_41_DSI S_9
TPCC.STOCK_42_DSI S_9
TPCC.STOCK_43_DSI S_10
TPCC.STOCK_44_DSI S_10
TPCC.STOCK_45_DSI S_11
TPCC.STOCK_46_DSI S_11
TPCC.STOCK_47_DSI S_12
TPCC.STOCK_48_DSI S_12
TPCC.STOCK_49_DSI S_1
TPCC.STOCK_50_DSI S_1
TPCC.STOCK_51_DSI S_2
TPCC.STOCK_52_DSI S_2
TPCC.STOCK_53_DSI S_3
TPCC.STOCK_54_DSI S_3
TPCC.STOCK_55_DSI S_4
TPCC.STOCK_56_DSI S_4
TPCC.STOCK_57_DSI S_5
TPCC.STOCK_58_DSI S_5
TPCC.STOCK_59_DSI S_6
TPCC.STOCK_60_DSI S_6
TPCC.STOCK_61_DSI S_7
TPCC.STOCK_62_DSI S_7
TPCC.STOCK_63_DSI S_8
TPCC.STOCK_64_DSI S_8
TPCC.STOCK_65_DSI S_9
TPCC.STOCK_66_DSI S_9
TPCC.STOCK_67_DSI S_10
TPCC.STOCK_68_DSI S_10
TPCC.STOCK_69_DSI S_11
TPCC.STOCK_70_DSI S_11
TPCC.STOCK_71_DSI S_12
TPCC.STOCK_72_DSI S_12
TPCC.STOCK_73_DSI S_1
TPCC.STOCK_74_DSI S_1
TPCC.STOCK_75_DSI S_2
TPCC.STOCK_76_DSI S_2
TPCC.STOCK_77_DSI S_7
TPCC.STOCK_78_DSI S_7
TPCC.STOCK_79_DSI S_10
TPCC.STOCK_80_DSI S_10
TPCC.STOCK_81_DSI S_3
TPCC.STOCK_82_DSI S_3
TPCC.STOCK_83_DSI S_6
TPCC.STOCK_84_DSI S_6
TPCC.ITEM_1_DSI I_1
TPCC.ORDERLIN_1_DSI OL_1
TPCC.ORDERLIN_2_DSI OL_2
TPCC.ORDERLIN_3_DSI OL_3
TPCC.ORDERLIN_4_DSI OL_4
TPCC.ORDERLIN_5_DSI OL_5
TPCC.ORDERLIN_6_DSI OL_6
TPCC.ORDERLIN_7_DSI OL_7
TPCC.ORDERLIN_8_DSI OL_8
TPCC.ORDERLIN_9_DSI OL_9
TPCC.ORDERLIN_10_DSI OL_10
TPCC.ORDERLIN_11_DSI OL_11
TPCC.ORDERLIN_12_DSI OL_12
TPCC.ORDERLIN_13_DSI OL_1
TPCC.ORDERLIN_14_DSI OL_2
TPCC.ORDERLIN_15_DSI OL_3
TPCC.ORDERLIN_16_DSI OL_4
TPCC.ORDERLIN_17_DSI OL_5
TPCC.ORDERLIN_18_DSI OL_6
TPCC.ORDERLIN_19_DSI OL_7
TPCC.ORDERLIN_20_DSI OL_8
TPCC.ORDERLIN_21_DSI OL_9
TPCC.ORDERLIN_22_DSI OL_10
TPCC.ORDERLIN_23_DSI OL_11
TPCC.ORDERLIN_24_DSI OL_12
TPCC.ORDERLIN_25_DSI OL_1
TPCC.ORDERLIN_26_DSI OL_2
TPCC.ORDERLIN_27_DSI OL_3
TPCC.ORDERLIN_28_DSI OL_4
TPCC.ORDERLIN_29_DSI OL_5
TPCC.ORDERLIN_30_DSI OL_6
TPCC.ORDERLIN_31_DSI OL_7
TPCC.ORDERLIN_32_DSI OL_8
TPCC.ORDERLIN_33_DSI OL_9
TPCC.ORDERLIN_34_DSI OL_10
TPCC.ORDERLIN_35_DSI OL_11
TPCC.ORDERLIN_36_DSI OL_12
TPCC.ORDERLIN_37_DSI OL_1
TPCC.ORDERLIN_38_DSI OL_2
TPCC.ORDERLIN_39_DSI OL_3
TPCC.ORDERLIN_40_DSI OL_4
TPCC.ORDERLIN_41_DSI OL_5
TPCC.ORDERLIN_42_DSI OL_6
TPCC.ORDERLIN_43_DSI OL_7
TPCC.ORDERLIN_44_DSI OL_8
TPCC.ORDERLIN_45_DSI OL_9
TPCC.ORDERLIN_46_DSI OL_10
TPCC.ORDERLIN_47_DSI OL_11
TPCC.ORDERLIN_48_DSI OL_12
TPCC.ORDERLIN_49_DSI OL_1
TPCC.ORDERLIN_50_DSI OL_2
TPCC.ORDERLIN_51_DSI OL_3
TPCC.ORDERLIN_52_DSI OL_4
TPCC.ORDERLIN_53_DSI OL_5
TPCC.ORDERLIN_54_DSI OL_6
TPCC.ORDERLIN_55_DSI OL_7
TPCC.ORDERLIN_56_DSI OL_8
TPCC.ORDERLIN_57_DSI OL_9
TPCC.ORDERLIN_58_DSI OL_10
TPCC.ORDERLIN_59_DSI OL_11
TPCC.ORDERLIN_60_DSI OL_12
TPCC.ORDERLIN_61_DSI OL_1
TPCC.ORDERLIN_62_DSI OL_2
TPCC.ORDERLIN_63_DSI OL_3
TPCC.ORDERLIN_64_DSI OL_4
TPCC.ORDERLIN_65_DSI OL_5
TPCC.ORDERLIN_66_DSI OL_6
TPCC.ORDERLIN_67_DSI OL_7
TPCC.ORDERLIN_68_DSI OL_8
TPCC.ORDERLIN_69_DSI OL_9
TPCC.ORDERLIN_70_DSI OL_10
TPCC.ORDERLIN_71_DSI OL_11
TPCC.ORDERLIN_72_DSI OL_12
TPCC.ORDERLIN_73_DSI OL_1
TPCC.ORDERLIN_74_DSI OL_2
TPCC.ORDERLIN_75_DSI OL_3
TPCC.ORDERLIN_76_DSI OL_4
TPCC.ORDERLIN_77_DSI OL_5
TPCC.ORDERLIN_78_DSI OL_6
TPCC.ORDERLIN_79_DSI OL_7
TPCC.ORDERLIN_80_DSI OL_8
TPCC.ORDERLIN_81_DSI OL_9
TPCC.ORDERLIN_82_DSI OL_10
TPCC.ORDERLIN_83_DSI OL_11
TPCC.ORDERLIN_84_DSI OL_12
TPCC.ORDERLIN_85_DSI OL_1
TPCC.ORDERLIN_86_DSI OL_2
TPCC.ORDERLIN_87_DSI OL_3
TPCC.ORDERLIN_88_DSI OL_4
TPCC.ORDERLIN_89_DSI OL_5
TPCC.ORDERLIN_90_DSI OL_6
TPCC.ORDERLIN_91_DSI OL_7
TPCC.ORDERLIN_92_DSI OL_8
TPCC.ORDERLIN_93_DSI OL_9
TPCC.ORDERLIN_94_DSI OL_10
TPCC.ORDERLIN_95_DSI OL_11
TPCC.ORDERLIN_96_DSI OL_12
TPCC.ORDERLIN_97_DSI OL_1
TPCC.ORDERLIN_98_DSI OL_2
TPCC.ORDERLIN_99_DSI OL_3
TPCC.ORDERLIN_100_DSI OL_4
TPCC.ORDERLIN_101_DSI OL_5
TPCC.ORDERLIN_102_DSI OL_6
TPCC.ORDERLIN_103_DSI OL_7
TPCC.ORDERLIN_104_DSI OL_8
TPCC.ORDERLIN_105_DSI OL_9
TPCC.ORDERLIN_106_DSI OL_10
TPCC.ORDERLIN_107_DSI OL_11
TPCC.ORDERLIN_108_DSI OL_12
TPCC.ORDERLIN_109_DSI OL_1
TPCC.ORDERLIN_110_DSI OL_2
TPCC.ORDERLIN_111_DSI OL_3
TPCC.ORDERLIN_112_DSI OL_4
TPCC.ORDERLIN_113_DSI OL_5
TPCC.ORDERLIN_114_DSI OL_6
TPCC.ORDERLIN_115_DSI OL_7
TPCC.ORDERLIN_116_DSI OL_8
TPCC.ORDERLIN_117_DSI OL_9
TPCC.ORDERLIN_118_DSI OL_10
TPCC.ORDERLIN_119_DSI OL_11
TPCC.ORDERLIN_120_DSI OL_12
TPCC.ORDERLIN_121_DSI OL_1
TPCC.ORDERLIN_122_DSI OL_2
TPCC.ORDERLIN_123_DSI OL_3
TPCC.ORDERLIN_124_DSI OL_4
TPCC.ORDERLIN_125_DSI OL_5
TPCC.ORDERLIN_126_DSI OL_6
TPCC.ORDERLIN_127_DSI OL_7
TPCC.ORDERLIN_128_DSI OL_8
TPCC.ORDERLIN_129_DSI OL_9
TPCC.ORDERLIN_130_DSI OL_10
TPCC.ORDERLIN_131_DSI OL_11
TPCC.ORDERLIN_132_DSI OL_12
TPCC.ORDERLIN_133_DSI OL_1
TPCC.ORDERLIN_134_DSI OL_2
TPCC.ORDERLIN_135_DSI OL_3
TPCC.ORDERLIN_136_DSI OL_4
TPCC.ORDERLIN_137_DSI OL_5
TPCC.ORDERLIN_138_DSI OL_6
TPCC.ORDERLIN_139_DSI OL_7
TPCC.ORDERLIN_140_DSI OL_8

TPCC.ORDERLIN_141_DSI OL_9
TPCC.ORDERLIN_142_DSI OL_10
TPCC.ORDERLIN_143_DSI OL_11
TPCC.ORDERLIN_144_DSI OL_12
TPCC.ORDERLIN_145_DSI OL_1
TPCC.ORDERLIN_146_DSI OL_2
TPCC.ORDERLIN_147_DSI OL_3
TPCC.ORDERLIN_148_DSI OL_4
TPCC.ORDERLIN_149_DSI OL_5
TPCC.ORDERLIN_150_DSI OL_6
TPCC.ORDERLIN_151_DSI OL_7
TPCC.ORDERLIN_152_DSI OL_8
TPCC.ORDERLIN_153_DSI OL_9
TPCC.ORDERLIN_154_DSI OL_10
TPCC.ORDERLIN_155_DSI OL_11
TPCC.ORDERLIN_156_DSI OL_12
TPCC.ORDERLIN_157_DSI OL_1
TPCC.ORDERLIN_158_DSI OL_2
TPCC.ORDERLIN_159_DSI OL_3
TPCC.ORDERLIN_160_DSI OL_4
TPCC.ORDERLIN_161_DSI OL_5
TPCC.ORDERLIN_162_DSI OL_6
TPCC.ORDERLIN_163_DSI OL_7
TPCC.ORDERLIN_164_DSI OL_8
TPCC.ORDERLIN_165_DSI OL_9
TPCC.ORDERLIN_166_DSI OL_10
TPCC.ORDERLIN_167_DSI OL_11
TPCC.ORDERLIN_168_DSI OL_12
TPCC.ORDERLIN_169_DSI OL_1
TPCC.ORDERLIN_170_DSI OL_2
TPCC.ORDERLIN_171_DSI OL_3
TPCC.ORDERLIN_172_DSI OL_4
TPCC.ORDERLIN_173_DSI OL_5
TPCC.ORDERLIN_174_DSI OL_6
TPCC.ORDERLIN_175_DSI OL_7
TPCC.ORDERLIN_176_DSI OL_8
TPCC.ORDERLIN_177_DSI OL_9
TPCC.ORDERLIN_178_DSI OL_10
TPCC.ORDERLIN_179_DSI OL_11
TPCC.ORDERLIN_180_DSI OL_12
TPCC.ORDERLIN_181_DSI OL_1
TPCC.ORDERLIN_182_DSI OL_2
TPCC.ORDERLIN_183_DSI OL_3
TPCC.ORDERLIN_184_DSI OL_4
TPCC.ORDERLIN_185_DSI OL_5
TPCC.ORDERLIN_186_DSI OL_6
TPCC.ORDERLIN_187_DSI OL_7
TPCC.ORDERLIN_188_DSI OL_8
TPCC.ORDERLIN_189_DSI OL_9
TPCC.ORDERLIN_190_DSI OL_10
TPCC.ORDERLIN_191_DSI OL_11
TPCC.ORDERLIN_192_DSI OL_12
TPCC.ORDERLIN_193_DSI OL_1
TPCC.ORDERLIN_194_DSI OL_2
TPCC.ORDERLIN_195_DSI OL_3
TPCC.ORDERLIN_196_DSI OL_4
TPCC.ORDERLIN_197_DSI OL_5
TPCC.ORDERLIN_198_DSI OL_6
TPCC.ORDERLIN_199_DSI OL_7
TPCC.ORDERLIN_200_DSI OL_8
TPCC.ORDERLIN_201_DSI OL_9
TPCC.ORDERLIN_202_DSI OL_10
TPCC.ORDERLIN_203_DSI OL_11
TPCC.ORDERLIN_204_DSI OL_12
TPCC.ORDERLIN_205_DSI OL_1
TPCC.ORDERLIN_206_DSI OL_2
TPCC.ORDERLIN_207_DSI OL_3
TPCC.ORDERLIN_208_DSI OL_4
TPCC.ORDERLIN_209_DSI OL_5
TPCC.ORDERLIN_210_DSI OL_6
TPCC.ORDERLIN_211_DSI OL_7
TPCC.ORDERLIN_212_DSI OL_8
TPCC.ORDERLIN_213_DSI OL_9
TPCC.ORDERLIN_214_DSI OL_10
TPCC.ORDERLIN_215_DSI OL_11
TPCC.ORDERLIN_216_DSI OL_12
TPCC.ORDERLIN_217_DSI OL_1
TPCC.ORDERLIN_218_DSI OL_2
TPCC.ORDERLIN_219_DSI OL_3
TPCC.ORDERLIN_220_DSI OL_4
TPCC.ORDERLIN_221_DSI OL_5
TPCC.ORDERLIN_222_DSI OL_6
TPCC.ORDERLIN_223_DSI OL_7
TPCC.ORDERLIN_224_DSI OL_8
TPCC.ORDERLIN_225_DSI OL_9
TPCC.ORDERLIN_226_DSI OL_10
TPCC.ORDERLIN_227_DSI OL_11
TPCC.ORDERLIN_228_DSI OL_12
TPCC.ORDERLIN_229_DSI OL_1
TPCC.ORDERLIN_230_DSI OL_2
TPCC.ORDERLIN_231_DSI OL_3
TPCC.ORDERLIN_232_DSI OL_4
TPCC.ORDERLIN_233_DSI OL_5
TPCC.ORDERLIN_234_DSI OL_6
TPCC.ORDERLIN_235_DSI OL_7
TPCC.ORDERLIN_236_DSI OL_8
TPCC.ORDERLIN_237_DSI OL_9
TPCC.ORDERLIN_238_DSI OL_10
TPCC.ORDERLIN_239_DSI OL_11
TPCC.ORDERLIN_240_DSI OL_12
TPCC.ORDERLIN_241_DSI OL_1
TPCC.ORDERLIN_242_DSI OL_2
TPCC.ORDERLIN_243_DSI OL_3
TPCC.ORDERLIN_244_DSI OL_4
TPCC.ORDERLIN_245_DSI OL_5
TPCC.ORDERLIN_246_DSI OL_6
TPCC.ORDERLIN_247_DSI OL_7
TPCC.ORDERLIN_248_DSI OL_8
TPCC.ORDERLIN_249_DSI OL_9
TPCC.ORDERLIN_250_DSI OL_10
TPCC.ORDERLIN_251_DSI OL_11
TPCC.ORDERLIN_252_DSI OL_12
TPCC.HISTORY_1_DSI H_1
TPCC.HISTORY_2_DSI H_1
TPCC.HISTORY_3_DSI H_1
TPCC.HISTORY_4_DSI H_1
TPCC.HISTORY_5_DSI H_1
TPCC.HISTORY_6_DSI H_1
TPCC.HISTORY_7_DSI H_2
TPCC.HISTORY_8_DSI H_2
TPCC.HISTORY_9_DSI H_2
TPCC.HISTORY_10_DSI H_2
TPCC.HISTORY_11_DSI H_2
TPCC.HISTORY_12_DSI H_2
TPCC.HISTORY_13_DSI H_3
TPCC.HISTORY_14_DSI H_3
TPCC.HISTORY_15_DSI H_3
TPCC.HISTORY_16_DSI H_3
TPCC.HISTORY_17_DSI H_3
TPCC.HISTORY_18_DSI H_3
TPCC.HISTORY_19_DSI H_4
TPCC.HISTORY_20_DSI H_4
TPCC.HISTORY_21_DSI H_4
TPCC.HISTORY_22_DSI H_4
TPCC.HISTORY_23_DSI H_4
TPCC.HISTORY_24_DSI H_4
TPCC.HISTORY_25_DSI H_5
TPCC.HISTORY_26_DSI H_5
TPCC.HISTORY_27_DSI H_5
TPCC.HISTORY_28_DSI H_5
TPCC.HISTORY_29_DSI H_5
TPCC.HISTORY_30_DSI H_5
TPCC.HISTORY_31_DSI H_6
TPCC.HISTORY_32_DSI H_6
TPCC.HISTORY_33_DSI H_6
TPCC.HISTORY_34_DSI H_6
TPCC.HISTORY_35_DSI H_6
TPCC.HISTORY_36_DSI H_6
TPCC.HISTORY_37_DSI H_7
TPCC.HISTORY_38_DSI H_7
TPCC.HISTORY_39_DSI H_7
TPCC.HISTORY_40_DSI H_7
TPCC.HISTORY_41_DSI H_7
TPCC.HISTORY_42_DSI H_7
TPCC.HISTORY_43_DSI H_8
TPCC.HISTORY_44_DSI H_8
TPCC.HISTORY_45_DSI H_8
TPCC.HISTORY_46_DSI H_8
TPCC.HISTORY_47_DSI H_8
TPCC.HISTORY_48_DSI H_8
TPCC.HISTORY_49_DSI H_9
TPCC.HISTORY_50_DSI H_9
TPCC.HISTORY_51_DSI H_9
TPCC.HISTORY_52_DSI H_9
TPCC.HISTORY_53_DSI H_9
TPCC.HISTORY_54_DSI H_9
TPCC.HISTORY_55_DSI H_10
TPCC.HISTORY_56_DSI H_10
TPCC.HISTORY_57_DSI H_10
TPCC.HISTORY_58_DSI H_10
TPCC.HISTORY_59_DSI H_10
TPCC.HISTORY_60_DSI H_10
TPCC.HISTORY_61_DSI H_11
TPCC.HISTORY_62_DSI H_11
TPCC.HISTORY_63_DSI H_11
TPCC.HISTORY_64_DSI H_11
TPCC.HISTORY_65_DSI H_11
TPCC.HISTORY_66_DSI H_11
TPCC.HISTORY_67_DSI H_12
TPCC.HISTORY_68_DSI H_12
TPCC.HISTORY_69_DSI H_12
TPCC.HISTORY_70_DSI H_12
TPCC.HISTORY_71_DSI H_12
TPCC.HISTORY_72_DSI H_12
TPCC.HISTORY_73_DSI H_1
TPCC.HISTORY_74_DSI H_1
TPCC.HISTORY_75_DSI H_1
TPCC.HISTORY_76_DSI H_1
TPCC.HISTORY_77_DSI H_1
TPCC.HISTORY_78_DSI H_1
TPCC.HISTORY_79_DSI H_2
TPCC.HISTORY_80_DSI H_2
TPCC.HISTORY_81_DSI H_2
TPCC.HISTORY_82_DSI H_2
TPCC.HISTORY_83_DSI H_2
TPCC.HISTORY_84_DSI H_2
TPCC.HISTORY_85_DSI H_3
TPCC.HISTORY_86_DSI H_3
TPCC.HISTORY_87_DSI H_3
TPCC.HISTORY_88_DSI H_3
TPCC.HISTORY_89_DSI H_3
TPCC.HISTORY_90_DSI H_3
TPCC.HISTORY_91_DSI H_4
TPCC.HISTORY_92_DSI H_4

TPCC.HISTORY_93_DSI H_4
TPCC.HISTORY_94_DSI H_4
TPCC.HISTORY_95_DSI H_4
TPCC.HISTORY_96_DSI H_4
TPCC.HISTORY_97_DSI H_5
TPCC.HISTORY_98_DSI H_5
TPCC.HISTORY_99_DSI H_5
TPCC.HISTORY_100_DSI H_5
TPCC.HISTORY_101_DSI H_5
TPCC.HISTORY_102_DSI H_5
TPCC.HISTORY_103_DSI H_6
TPCC.HISTORY_104_DSI H_6
TPCC.HISTORY_105_DSI H_6
TPCC.HISTORY_106_DSI H_6
TPCC.HISTORY_107_DSI H_6
TPCC.HISTORY_108_DSI H_6
TPCC.HISTORY_109_DSI H_7
TPCC.HISTORY_110_DSI H_7
TPCC.HISTORY_111_DSI H_7
TPCC.HISTORY_112_DSI H_7
TPCC.HISTORY_113_DSI H_7
TPCC.HISTORY_114_DSI H_7
TPCC.HISTORY_115_DSI H_8
TPCC.HISTORY_116_DSI H_8
TPCC.HISTORY_117_DSI H_8
TPCC.HISTORY_118_DSI H_8
TPCC.HISTORY_119_DSI H_8
TPCC.HISTORY_120_DSI H_8
TPCC.HISTORY_121_DSI H_9
TPCC.HISTORY_122_DSI H_9
TPCC.HISTORY_123_DSI H_9
TPCC.HISTORY_124_DSI H_9
TPCC.HISTORY_125_DSI H_9
TPCC.HISTORY_126_DSI H_9
TPCC.HISTORY_127_DSI H_10
TPCC.HISTORY_128_DSI H_10
TPCC.HISTORY_129_DSI H_10
TPCC.HISTORY_130_DSI H_10
TPCC.HISTORY_131_DSI H_10
TPCC.HISTORY_132_DSI H_10
TPCC.HISTORY_133_DSI H_11
TPCC.HISTORY_134_DSI H_11
TPCC.HISTORY_135_DSI H_11
TPCC.HISTORY_136_DSI H_11
TPCC.HISTORY_137_DSI H_11
TPCC.HISTORY_138_DSI H_11
TPCC.HISTORY_139_DSI H_12
TPCC.HISTORY_140_DSI H_12
TPCC.HISTORY_141_DSI H_12
TPCC.HISTORY_142_DSI H_12
TPCC.HISTORY_143_DSI H_12
TPCC.HISTORY_144_DSI H_12
TPCC.HISTORY_145_DSI H_1
TPCC.HISTORY_146_DSI H_1
TPCC.HISTORY_147_DSI H_1
TPCC.HISTORY_148_DSI H_1
TPCC.HISTORY_149_DSI H_1
TPCC.HISTORY_150_DSI H_1
TPCC.HISTORY_151_DSI H_2
TPCC.HISTORY_152_DSI H_2
TPCC.HISTORY_153_DSI H_2
TPCC.HISTORY_154_DSI H_2
TPCC.HISTORY_155_DSI H_2
TPCC.HISTORY_156_DSI H_2
TPCC.HISTORY_157_DSI H_3
TPCC.HISTORY_158_DSI H_3
TPCC.HISTORY_159_DSI H_3
TPCC.HISTORY_160_DSI H_3
TPCC.HISTORY_161_DSI H_3
TPCC.HISTORY_162_DSI H_3
TPCC.HISTORY_163_DSI H_4
TPCC.HISTORY_164_DSI H_4
TPCC.HISTORY_165_DSI H_4
TPCC.HISTORY_166_DSI H_4
TPCC.HISTORY_167_DSI H_4
TPCC.HISTORY_168_DSI H_4
TPCC.HISTORY_169_DSI H_5
TPCC.HISTORY_170_DSI H_5
TPCC.HISTORY_171_DSI H_5
TPCC.HISTORY_172_DSI H_5
TPCC.HISTORY_173_DSI H_5
TPCC.HISTORY_174_DSI H_5
TPCC.HISTORY_175_DSI H_6
TPCC.HISTORY_176_DSI H_6
TPCC.HISTORY_177_DSI H_6
TPCC.HISTORY_178_DSI H_6
TPCC.HISTORY_179_DSI H_6
TPCC.HISTORY_180_DSI H_6
TPCC.HISTORY_181_DSI H_7
TPCC.HISTORY_182_DSI H_7
TPCC.HISTORY_183_DSI H_7
TPCC.HISTORY_184_DSI H_7
TPCC.HISTORY_185_DSI H_7
TPCC.HISTORY_186_DSI H_7
TPCC.HISTORY_187_DSI H_8
TPCC.HISTORY_188_DSI H_8
TPCC.HISTORY_189_DSI H_8
TPCC.HISTORY_190_DSI H_8
TPCC.HISTORY_191_DSI H_8
TPCC.HISTORY_192_DSI H_8
TPCC.HISTORY_193_DSI H_9
TPCC.HISTORY_194_DSI H_9
TPCC.HISTORY_195_DSI H_9
TPCC.HISTORY_196_DSI H_9
TPCC.HISTORY_197_DSI H_9
TPCC.HISTORY_198_DSI H_9
TPCC.HISTORY_199_DSI H_10
TPCC.HISTORY_200_DSI H_10
TPCC.HISTORY_201_DSI H_10
TPCC.HISTORY_202_DSI H_10
TPCC.HISTORY_203_DSI H_10
TPCC.HISTORY_204_DSI H_10
TPCC.HISTORY_205_DSI H_11
TPCC.HISTORY_206_DSI H_11
TPCC.HISTORY_207_DSI H_11
TPCC.HISTORY_208_DSI H_11
TPCC.HISTORY_209_DSI H_11
TPCC.HISTORY_210_DSI H_11
TPCC.HISTORY_211_DSI H_12
TPCC.HISTORY_212_DSI H_12
TPCC.HISTORY_213_DSI H_12
TPCC.HISTORY_214_DSI H_12
TPCC.HISTORY_215_DSI H_12
TPCC.HISTORY_216_DSI H_12
TPCC.HISTORY_217_DSI H_1
TPCC.HISTORY_218_DSI H_1
TPCC.HISTORY_219_DSI H_1
TPCC.HISTORY_220_DSI H_1
TPCC.HISTORY_221_DSI H_1
TPCC.HISTORY_222_DSI H_1
TPCC.HISTORY_223_DSI H_2
TPCC.HISTORY_224_DSI H_2
TPCC.HISTORY_225_DSI H_2
TPCC.HISTORY_226_DSI H_2
TPCC.HISTORY_227_DSI H_2
TPCC.HISTORY_228_DSI H_2
TPCC.HISTORY_229_DSI H_3
TPCC.HISTORY_230_DSI H_3
TPCC.HISTORY_231_DSI H_3
TPCC.HISTORY_232_DSI H_3
TPCC.HISTORY_233_DSI H_3
TPCC.HISTORY_234_DSI H_3
TPCC.HISTORY_235_DSI H_4
TPCC.HISTORY_236_DSI H_4
TPCC.HISTORY_237_DSI H_4
TPCC.HISTORY_238_DSI H_4
TPCC.HISTORY_239_DSI H_4
TPCC.HISTORY_240_DSI H_4
TPCC.HISTORY_241_DSI H_5
TPCC.HISTORY_242_DSI H_5
TPCC.HISTORY_243_DSI H_5
TPCC.HISTORY_244_DSI H_5
TPCC.HISTORY_245_DSI H_5
TPCC.HISTORY_246_DSI H_5
TPCC.HISTORY_247_DSI H_6
TPCC.HISTORY_248_DSI H_6
TPCC.HISTORY_249_DSI H_6
TPCC.HISTORY_250_DSI H_6
TPCC.HISTORY_251_DSI H_6
TPCC.HISTORY_252_DSI H_6

File: crbuf.aa

W_1 1K 396 100 100 1 2147483646 fixed
W_2 1K 396 100 100 1 2147483646 fixed
W_3 1K 396 100 100 1 2147483646 fixed
W_4 1K 396 100 100 1 2147483646 fixed
W_5 1K 396 100 100 1 2147483646 fixed
W_6 1K 396 100 100 1 2147483646 fixed
W_7 1K 396 100 100 1 2147483646 fixed

D_1 1K 3960 100 100 1 2147483646 fixed
D_2 1K 3960 100 100 1 2147483646 fixed
D_3 1K 3960 100 100 1 2147483646 fixed
D_4 1K 3960 100 100 1 2147483646 fixed
D_5 1K 3960 100 100 1 2147483646 fixed
D_6 1K 3960 100 100 1 2147483646 fixed
D_7 1K 3960 100 100 1 2147483646 fixed

C_1 1K 3 90 80 1 2147483646 fixed
C_2 1K 3 90 80 1 2147483646 fixed
C_3 1K 3 90 80 1 2147483646 fixed
C_4 1K 3 90 80 1 2147483646 fixed
C_5 1K 3 90 80 1 2147483646 fixed
C_6 1K 3 90 80 1 2147483646 fixed
C_7 1K 3 90 80 1 2147483646 fixed
C_8 1K 3 90 80 1 2147483646 fixed
C_9 1K 3 90 80 1 2147483646 fixed
C_10 1K 3 90 80 1 2147483646 fixed
C_11 1K 3 90 80 1 2147483646 fixed
C_12 1K 3 90 80 1 2147483646 fixed

C_IX_1 32K 26 100 100 1 2147483646 fixed
C_IX_2 32K 26 100 100 1 2147483646 fixed
C_IX_3 32K 20 100 100 1 2147483646 fixed
C_IX_4 32K 20 100 100 1 2147483646 fixed
C_IX_5 32K 22 100 100 1 2147483646 fixed

```
C_IX_6      32K   22     100   100   1
2147483646 fixed
C_IX_7      32K   20     100   100   1
2147483646 fixed
C_IX_8      32K   20     100   100   1
2147483646 fixed
C_IX_9      32K   20     100   100   1
2147483646 fixed
C_IX_10     32K   20     100   100   1
2147483646 fixed
C_IX_11     32K   18     100   100   1
2147483646 fixed
C_IX_12     32K   18     100   100   1
2147483646 fixed

O_1  1K    3      90    80    1 2147483646 fixed
O_2  1K    3      90    80    1 2147483646 fixed
O_3  1K    3      90    80    1 2147483646 fixed
O_4  1K    3      90    80    1 2147483646 fixed
O_5  1K    3      90    80    1 2147483646 fixed
O_6  1K    3      90    80    1 2147483646 fixed
O_7  1K    3      90    80    1 2147483646 fixed
O_8  1K    3      90    80    1 2147483646 fixed
O_9  1K    3      90    80    1 2147483646 fixed
O_10 1K    3      90    80    1 2147483646 fixed
O_11 1K    3      90    80    1 2147483646 fixed
O_12 1K    3      90    80    1 2147483646 fixed

O_IX_1      32K   260    100   100   1
2147483646 fixed
O_IX_2      32K   260    100   100   1
2147483646 fixed
O_IX_3      32K   200    100   100   1
2147483646 fixed
O_IX_4      32K   200    100   100   1
2147483646 fixed
O_IX_5      32K   220    100   100   1
2147483646 fixed
O_IX_6      32K   220    100   100   1
2147483646 fixed
O_IX_7      32K   200    100   100   1
2147483646 fixed
O_IX_8      32K   200    100   100   1
2147483646 fixed
O_IX_9      32K   200    100   100   1
2147483646 fixed
O_IX_10     32K   200    100   100   1
2147483646 fixed
O_IX_11     32K   180    100   100   1
2147483646 fixed
O_IX_12     32K   180    100   100   1
2147483646 fixed

NO_11K    3      90    80    1 2147483646 fixed
NO_21K    3      90    80    1 2147483646 fixed
NO_31K    3      90    80    1 2147483646 fixed
NO_41K    3      90    80    1 2147483646 fixed
NO_51K    3      90    80    1 2147483646 fixed
NO_61K    3      90    80    1 2147483646 fixed
NO_71K    3      90    80    1 2147483646 fixed
NO_81K    3      90    80    1 2147483646 fixed
NO_91K    3      90    80    1 2147483646 fixed
NO_10       1K    3      90    80    1
2147483646 fixed
NO_11       1K    3      90    80    1
2147483646 fixed
NO_12       1K    3      90    80    1
2147483646 fixed

NO_IX_1     32K   21     100   100   1
2147483646 fixed
NO_IX_2     32K   21     100   100   1
2147483646 fixed
NO_IX_3     32K   21     100   100   1
2147483646 fixed
NO_IX_4     32K   21     100   100   1
2147483646 fixed
NO_IX_5     32K   21     100   100   1
2147483646 fixed
NO_IX_6     32K   21     100   100   1
2147483646 fixed
NO_IX_7     32K   21     100   100   1
2147483646 fixed
NO_IX_8     32K   21     100   100   1
2147483646 fixed
NO_IX_9     32K   21     100   100   1
2147483646 fixed
NO_IX_10    32K   21     100   100   1
2147483646 fixed
NO_IX_11    32K   21     100   100   1
2147483646 fixed
NO_IX_12    32K   21     100   100   1
2147483646 fixed

I_1    1K     14300       100   100   1
2147483646 fixed

S_1 1K 3 90 80 1 2147483646 fixed
S_2 1K 3 90 80 1 2147483646 fixed
S_3 1K 3 90 80 1 2147483646 fixed
S_4 1K 3 90 80 1 2147483646 fixed
S_5 1K 3 90 80 1 2147483646 fixed
S_6 1K 3 90 80 1 2147483646 fixed
S_7 1K 3 90 80 1 2147483646 fixed
S_8 1K 3 90 80 1 2147483646 fixed
S_9 1K 3 90 80 1 2147483646 fixed
S_10 1K 3 90 80 1 2147483646 fixed
S_11 1K 3 90 80 1 2147483646 fixed
S_12 1K 3 90 80 1 2147483646 fixed
```

File: fssqlenv

```
;
; All Rights Reserved, Copyright(c) FUJITSU 1993,
1994, 1995
; All Rights Reserved, Copyright(c) PFU     1993,
1994, 1995
;
;     :
;
;
;           :          ';'
;           :
;           : 1                    1024
;
;;;
MAX_CONNECT_TCP=(260)
;MAX_CONNECT_SYS=(250)
;MAX_CONNECT_SYS=(470)
MAX_CONNECT_SYS=(510)
; rdbsysconfig -> RDBCNTNUM
JOIN_RULE=(F,3)
COMMUNICATION_BUFFER=(8)
;INCLUSION_TYPE=NEW
;MAX_PARALLEL=20
RDB2_TCP_LEVEL=(LEVEL2)
;SOC_SELECT_TIME=(20000)
SOC_SELECT_TIME=(5000)
SOC_WRITE_SIZE=(8)
;DEBUG_INFO = ON,2084
```

File: rdbbuf

```
#
# All Rights Reserved, Copyright(c) FUJITSU
1993, 1994, 1995
# All Rights Reserved, Copyright(c) PFU     1993,
1994, 1995
#
#     :
#
#
#           : '#'
#           :
#           : 1                    1024
#
#    <<        >>
#                    =
#
########################################
##################################nrk
BUFFER1K  = 64
BUFFER2K  = 64
BUFFER4K  = 512
BUFFER8K  = 1024
BUFFER16K = 32
BUFFER32K = 32
```

File: rdbcwbuf.wk

```
ITEM,STOCK
100,6519,1,4,0,0,4,4,4,1,8,1,24,2,32,5,2,3,34,1,50,
1,1,0,4,4,100000,100000
100,462,0,1,0,0,4,5,2,2,1,0,5,2,2772,2772,0,4,4,1
00000,100000
```

File: rdbpool

```
#
# All Rights Reserved, Copyright(c) FUJITSU
1993, 1994, 1995, 1996
# All Rights Reserved, Copyright(c) PFU     1993,
1994, 1995, 1996
#
#     :
#
#
#           : '#'
#           :
#           : 1                    1024
#
#    <<        >>
#                    =       ,       ,
#
########################################
##################################nrk
```

```
# system
#         =                       #
#-------------------------------------------------------------------
-----------
ARC_ALCT    = 0     ,1     ,1024000000  #508
BCM_BPC     = 124928 ,4096  ,1024000000  #508
#BCM_EEXT    = 294912 ,8192  ,1024000000
#1532
BCM_EEXT    = 310272 ,8192  ,1024000000
#1532
BCM_ESUB    = 351232 ,1024  ,1024000000  #60
BCM_GPCT    = 4096   ,4096  ,1024000000  #508
#BCM_IOPROC  = 71680  ,4096  ,1024000000
#380
BCM_IOPROC  = 78848  ,4096  ,1024000000
#380
BCM_LOGAREA = 0      ,1     ,1024000000  #1020
BCM_LOGLIST = 0      ,1     ,1024000000  #252
#BCM_PGC     = 176128 ,4096  ,1024000000
#1020
BCM_PGC     = 249856 ,4096  ,1024000000
#1020
BCM_WKACC   = 0      ,1024  ,1024000000  #252
BCM_WKDMON  = 256    ,1024  ,1024000000
#252
BCM_WKSPC   = 0      ,1024  ,1024000000  #60
BCM_WKSSPC  = 0      ,1024  ,1024000000  #124
#CCR_COMINF  = 2121728 ,16384 ,1024000000
#3836(0xefc)
CCR_COMINF  = 2629632 ,16384 ,1024000000
#3836(0xefc)
#CCR_FGRP    = 43008  ,4096  ,1024000000
#252
CCR_FGRP    = 63488 ,4096  ,1024000000  #252
#CCR_IDT     = 849920 ,4096  ,1024000000
#912(0x390)
CCR_IDT     = 848896 ,4096  ,1024000000
#912(0x390)
#CCR_KAIOCB  = 184320  ,1024  ,1024000000
#92
CCR_KAIOCB  = 271360  ,1024  ,1024000000
#92
#CCR_LWPIDT  = 157696 ,4096  ,1024000000
#112(0x70)
CCR_LWPIDT  = 15360 ,4096  ,1024000000
#112(0x70)
#CCR_POLMCTL = 4184064 ,16384 ,1024000000
#4336(0x10+0x50*(42+12))
CCR_POLMCTL = 2048000 ,16384 ,1024000000
#4336(0x10+0x50*(42+12))
CCR_SANQUE  = 29696  ,1024  ,1024000000
#64
CCR_USRCON  = 412672 ,4096  ,1024000000
#448(0x1c0)
#CCR_USRSTK  =
25600000 ,65544 ,1024000000  #65536
CCR_USRSTK  = 43523072 ,65544 ,1024000000
#65536
#CCR_WLIST   = 17408  ,1024  ,1024000000
#28(0x1c)
CCR_WLIST   = 22528  ,1024  ,1024000000
#28(0x1c)
#CCR_WPID    = 276480  ,1024  ,1024000000
#60
CCR_WPID    = 306288  ,1024  ,1024000000  #60
DSM_DSAH    = 786432 ,4096  ,1024000000
#2044
#DSM_DSAP    =
45056000 ,1024000  ,1024000000  #124
#DSM_DSAP    =
34816000 ,1024000  ,1024000000  #124
#DSM_DSAP    =
32768000 ,1024000  ,1024000000  #124
DSM_DSAP    =
67584000 ,1024000  ,1024000000  #124
#DSM_DSIL    = 2668544 ,1024  ,1024000000
#60
DSM_DSIL    = 3085312 ,1024  ,1024000000  #60
DSM_DSVP    = 7168   ,1024  ,1024000000  #60
#DSM_DSVQ    = 50176  ,1024  ,1024000000
#92
DSM_DSVQ    = 61440  ,1024  ,1024000000  #92
DSM_DSWH    = 0      ,4096  ,1024000000  #2044
DSM_DSWP    = 0      ,1024  ,1024000000  #124
#DSM_DUSI    = 12288  ,1024  ,1024000000  #60
DSM_DUSI    = 16384  ,1024  ,1024000000  #60
DSM_DWFL    = 0      ,1024  ,1024000000  #60
DSM_DWUI    = 0      ,1024  ,1024000000  #60
#LCM_LOGCNTL = 47104  ,4096  ,1024000000
#252
LCM_LOGCNTL = 63488  ,4096  ,1024000000
#252
SCI_CMD     = 73728  ,4096  ,1024000000  #508
SCI_CONBF   = 4096   ,4096  ,1024000000  #508
#SSV_IINF    = 20480  ,1024  ,1024000000  #124
SSV_IINF    = 31774  ,1024  ,1024000000  #124
#TCM_TRAN    = 43008  ,4096  ,1024000000
#252
TCM_TRAN    = 63488  ,4096  ,1024000000  #252
UTY_UNQUE   = 256    ,1024  ,1024000000  #172
UTY_UNDB    = 0      ,4096  ,1024000000  #508
UTY_UNDSI   = 0      ,1024  ,1024000000  #124
XCM_KHASH   = 0      ,4096  ,1024000000  #1036
XCM_KMEM    = 0      ,4096  ,1024000000  #2044
XCM_KQUE    = 0      ,1024  ,1024000000  #28
XCM_KTERM   = 0      ,1024  ,1024000000  #28
#XCM_LOCK    = 763904 ,10240  ,1024000000
#60
XCM_LOCK    = 1024000 ,10240  ,1024000000
#60
XCM_LPHASH  = 6784000 ,4096  ,1024000000
#1028
XCM_NLOWN   = 5120   ,1024  ,1024000000  #28
#XCM_NLQUE   =
29278208 ,10240  ,1024000000  #60
#XCM_NLRSC   = 1124352 ,1024  ,1024000000
#252
XCM_NLQUE   = 10240 ,10240  ,1024000000
#60
XCM_NLRSC   = 10240 ,1024  ,1024000000
#252
XCM_OWNER   = 22528  ,1024  ,1024000000
#124
#XCM_QUE     = 3136512 ,102400  ,1024000000
#124
XCM_QUE     = 3379200 ,102400  ,1024000000
#124
XCM_TTERM   = 841728 ,1024  ,1024000000
#44
XCM_WQUE_S  = 0      ,1024  ,1024000000  #76
XCM_RSC_S   = 0      ,1024  ,1024000000  #60
#-------------------------------------------------------------------
-----------
# group
#         =                     #
#-------------------------------------------------------------------
-----------
BCM_DFPOOL_G = 128    ,1024  ,1024000000
#124
BCM_DPCT_G   = 64     ,1024  ,1024000000  #60
#CCR_GCOMINF  = 32768  ,16384 ,1024000000
#3836(0xefc)
CCR_GCOMINF  = 8192  ,16384 ,1024000000
#3836(0xefc)
XCM_BITMAP_G = 96     ,1024  ,1024000000
#92
XCM_BITMNG_G = 64     ,1024  ,1024000000
#60
#XCM_RSC_G    = 64     ,1024  ,1024000000  #60
XCM_RSC_G    = 3520    ,1024  ,1024000000
#60
#XCM_WQUE_G   = 80     ,1024  ,1024000000
#76
XCM_WQUE_G   = 20000   ,1024  ,1024000000
#76
#-------------------------------------------------------------------
-----------
# local
#         =                     #
#-------------------------------------------------------------------
-----------
BCM_DFPOOL  = 128    ,1024  ,1024000000
#124
BCM_DPCT    = 64     ,1024  ,1024000000  #60
#BCM_LPCT    = 64     ,1024  ,1024000000  #60
BCM_LPCT    = 960    ,1024  ,1024000000  #60
#BCM_LPG     = 256    ,1024  ,1024000000  #252
BCM_LPG     = 256    ,270000  ,1024000000
#252
BCM_PFT     = 256    ,67000  ,1024000000  #252
CCR_LCOMINF = 4120   ,307200 ,1024000000
#3836(0xefc)
#CCR_LCOMINF = 8192   ,40960 ,1024000000
#3836(0xefc)
DSM_DDSF    = 256    ,1024  ,1024000000  #252
##DSM_DDSF    = 411136  ,1024  ,1024000000
#252
DSM_DESF    = 256    ,1024  ,1024000000  #252
##DSM_DESF    = 736000    ,1024  ,1024000000
#252
SAP_KEY     = 4096   ,16384 ,1024000000  #4092
SCI_CS      = 0      ,1     ,1024000000  #124
#XCM_BITMAP  = 96     ,1024  ,1024000000  #92
XCM_BITMAP  = 96     ,20000  ,1024000000  #92
XCM_BITMNG  = 64     ,2400  ,1024000000  #60
XCM_RSC     = 64     ,1024  ,1024000000  #60
XCM_WQUE    = 80     ,1024  ,1024000000  #76
XCM_THASH   = 48     ,1400   ,1024000000  #44
#XCM_TQUE    = 80     ,1024  ,1024000000  #76
XCM_TQUE    = 400    ,1024  ,1024000000  #76
```

File: Rdbstart

```
#!/usr/bin/sh -xf
RDBDB=TPCC

#RCV_LOGBUFNUM=1024; export
RCV_LOGBUFNUM   #for down rcv
#RCV_MULTINUM=32; export RCV_MULTINUM
#for down rcv
```

```
#SEINO_DSM_PRT=2; export SEINO_DSM_PRT

RDBBUFCPU=1
export RDBBUFCPU

RDB_UNBALANCED_CPUBIND='88 65 89 71 87
10 88 68' #2640wh for online
export RDB_UNBALANCED_CPUBIND

RDBFOPENMAX2=30
export RDBFOPENMAX2

RDBBUFPATH=/rdbptc/tpcc80/tpcc/b-
onsrc/rdbcwbuf.wk
export RDBBUFPATH

RDB_LOCAL_LP_ADDR=0x3000000
export RDB_LOCAL_LP_ADDR
#RDB_LOCAL_LP_SIZE=896
RDB_LOCAL_LP_SIZE=800
export RDB_LOCAL_LP_SIZE

#RDB_LOCAL_LP_ADDR2=0x3b000000
#980601
RDB_LOCAL_LP_ADDR2=0x35000000  #980616
export RDB_LOCAL_LP_ADDR2

RDB_LOCAL_LP_SIZE2=552     #980614
export RDB_LOCAL_LP_SIZE2

#RDBINFKB=1
#export RDBINFKB

date; rdbstart

rdbcrbufcw cw_env.comp

date

#csh sh.crbuf.mk.7sdp.MBS.oza
sh sh.crbuf

timex rdbcrbf -f crbuf.aa  2>crbf.wk.out1  #WH
grep -v qdg02630 crbf.wk.out1

sh conbf.sh

date
sar -r 1 1
swap -s


#ITEM
RDBMSG=E
export RDBMSG

wupi.sh #(            ITEM READ)

sar -r 1 1
swap -s

crbfadd.sh
date

cwenvchk

echo "RDBSTART env" >
res.doc/RDBSTART.env
env >> res.doc/RDBSTART.env
cwenvchk >>res.doc/RDBSTART.env
crbfadd.sh >>res.doc/RDBSTART.env


rdbpid
set +x
if [ `rdbpid` != RDBII                              ]
then
        echo rdblog -V -a
        rdblog -V -a
fi
```

File: Rdbstop

```
#!/bin/csh -f
rm -rf  /tmp/ipcrm.csh
set rdbpidchk = `rdbpid`
if( $rdbpidchk == "RDBII                              " ) exit
set varbose
timex rdbrcp

#rdblog -R -a
#rdblog -R -a
#rdblog -R -a
#rdblog -R -a
#rdblog -R -a
#rdblog -R -a
#rdblog -S -a
#rdblog -R -a
#rdblog -V -a
date;
sleep 2
echo "   RDBII         "
#lcd 0 2 "RDB: closing now"
sleep 2
timex rdbstop
set rdbcheck = $status

#lcd 0 2 "(Ext.4197 Ozawa)"


if( $rdbcheck == 1 ) then
        echo "RDBSTOP:     : RDBII
ipcrm         "
        exit
endif

sleep 3

set rdbpidchk = `rdbpid`

if( $rdbpidchk == "RDBII                              " ) then
        date;
        echo "            shm / semaphores        "
        echo "<<
>>===================================
============================="
        ipcs
        printf "ipcrm_s " > /tmp/ipcrm.csh
        ipcs | grep seino | nawk '{ printf "-%s %s
",$1,$2 }' >> /tmp/ipcrm.csh
        printf "\n"  >> /tmp/ipcrm.csh
        chmod a+x /tmp/ipcrm.csh
        #echo "please  /tmp/ipcrm.csh"
        /tmp/ipcrm.csh
        echo "<<
>>===================================
============================="
        ipcs
        exit
else
        echo "RDBSTOP:     : RDBII
ipcrm         "
endif
```

File: rdbsysconfig

```
#
# All Rights Reserved, Copyright(c) FUJITSU
1993, 1994, 1995
# All Rights Reserved, Copyright(c) PFU     1993,
1994, 1995
#
#      : RDBII
#
#
#             : '#'
#             :
#             : 1                              1024
#             :                        1      214748367
#
#     <<          >>
#                         = [      1] , [      2] ,
#
############################################
#################################nrk
RDBDIRSPACE1=/home/RDBDIRE
RDBDIRSPACE2=/home/RDBDIRE
#RDBLOG=512, 512
RDBLOG=1024, 1024

RDBCORE=/rdbptc/RDBCORE
RDBCNTNUM=50          #for cl-sv #fssqlenv ->
MAX_CONECT_SYS
#RDBCNTNUM=470           #for batch #fssqlenv -
> MAX_CONECT_SYS
##RDBCNTNUM=475          #for batch #fssqlenv
-> MAX_CONECT_SYS
#NG-RDBCNTNUM=505 NG
RDBPRJCODE=0xdb

RDBSYSBUF=/rdbptc/tpcc80/SYS
RDBSQLENV=/rdbptc/tpcc80/SYS/fssqlenv
RDBLOGMANAGE=/rdbptc/tpcc80/SYS
RDBPOOLCFG=/rdbptc/tpcc80/SYS

#-----------------------------------------------------------------
#RDBFIXBUFMEM=80             #LOAD
#RDBFIXBUFMEMADDR=0x36000000

#old  #RDBFIXBUFMEM=2272           #sokutei
old

#2508WH RDBFIXBUFMEM=2288
#sokutei old
#RDBFIXBUFMEM=2256           # sokutei
(2772WH 990604)
RDBFIXBUFMEM=2352           # sokutei
(2772WH 990604)

#RDBFIXBUFMEMADDR=0x5e000000  #
```

```
RDBFIXBUFMEMADDR=0x59000000  #
#2508WHnotoki
#RDBFIXBUFMEMADDR=0x5c000000  #
#old  #RDBFIXBUFMEMADDR=0x5d000000  #old
#-------------------------------------------------------
RDBEXTMEMADDR=0xed000000
RDBEXTMEM=8192

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

RDBKCHKSKIPCNT=200    #        IO
96.10.05
#RDBKCHKSKIPCNT=300    #        IO
96.10.05
RDBKTAJUUDOSDP = 2016
RDBKLISTNUMSDP = 5
RDBKCATENUMSDP=252
#RDBKAIOREP = 6
RDBKAIOREP = 12  #980531
#RDBKAIOREP = 18  #980531
##RDBKAIOSAV = 30
RDBKAIOYLD = 0

RDBKAIOCNT    = yes
RDBKAIOD9F    = yes
RDBKAIOSELFWAIT    = yes
RDBKAIODSP    = yes
#RDBIOERRDOWN = yes
########################################
##############################nrk
##RDBVER:
UXP/DS_RDBII_V20L21_3/22_version
########################################
##################################
#RDBSDPCPU =  0,1,2,3,4,5,6,7   ## SDP
CPU
#RDBSDPCPU =  0,1,2,3,4,6   ## SDP
CPU
#RDBSDPCPU =  0,2,4,6   ## SDP              CPU
#---
RDBSDPCPU =  0,1,2,3,4,6,7   ## SDP
CPU

RDBCCRDMCPU=5               ##       daemon
CPU
RDBRECEPCPU=5
#---
RDBSORTCPU=5
RDBTCPIPCPU=5
RDBALFCPU=5   #990610 5 --> 1 --> 5
#RDBDBSCPU=1,2,3,4,5,6,7          ##
RDBDBSCPU=1          ##
#
RDBDIRCPU=5   #990610 5 --> none-->5
#---
RDBIOCPU=5
RDBTLFCPU=1 #5   #990610 5 --> 1 -->5
#---
RDBWKSCPU=5



####I/O
RDBREADUNC = NO            # SAP
6/28

RDBDBSNUM+ = 350          #      /

RDBMAXLWP =  100         #98.11.11 oza
#RDBMAXDBIO = 20          # 20
I/O
RDBMAXDBIO = 12 #14ng  #12ok #5NG #10  #
990309 oza
RDBMAXRCPIO = 13 #15?  #13ok #6NG #11 #
old=20 RCP                I/O
RDBNEWPAGE = 1            # LRU
#RDBNEWPAGE = 2            # LRU
RDBANTIQUEPAGE = 4 # LRU  4             ,
RDBIOUNITNUM = 1,1
RDBSORTUNITNUM = 1,1

##LOG                    ( GC    =6    =30ms
>2 BUF=32 )
#RDBLOGAIONUM=32                #IO buffer
RDBLOGAIONUM=128 #90           #IO buffer
RDBLOGBIONUM=256 #90           #
RDBLOGIOSLEEP=10               #SLTRNUM
tran  write             sleep
RDBLOGSLTRNUM=2                          #
tran          write
RDBLOGGRCOMMIT=4
        #SLEEP                 tran           nosleep

#RDBKTAJUUDOSDP=5
#RDBKCATENUMSDP=5
#RDBKCATENUMSDP=60


##SLK/LWP          tuning (0418           )
RDBSLKLOOP=10        #   100steps
#RDBLWPLOOP=100                  #   10steps
MIPS=10 (1min.=600Msteps)
RDBSEMMODE=IPC   #mutex      IPC

##my_mutex      flag
#RDBDBGSLKCNT=yes,yes
#RDBDBGSLKCNT=yes
#RDBDBGMUTCNT=yes
#RDBDBGWPCCNT = yes
#RDBDBGWAITPOS = yes
#RDBKAIOCNT    = yes

##mutex                  BUSY
#RDBMUTLOOP=100

##
mutex
.
#RDBMUTGIVE=YES
RDBIXSECDWD=YES      #

RDBSDPLDBALMODE = 2



File:  rdbsysconfig.LOAD

#
# All Rights Reserved, Copyright(c) FUJITSU
1993, 1994, 1995
# All Rights Reserved, Copyright(c) PFU     1993,
1994, 1995
#
#     : RDBII
#
#
#             : '#'
#             :
#             : 1                          1024
#             :                  1      214748367
#
#    <<         >>
#                    = [      1] , [      2] ,
#
########################################
##############################nrk
RDBDIRSPACE1=/home/RDBDIRE
RDBDIRSPACE2=/home/RDBDIRE
RDBLOG=512, 512
#RDBLOG=256, 128
#RDBLOG=1024, 512
RDBCORE=/rdbptc/RDBCORE
#RDBCNTNUM=250          #for cl-sv #fssqlenv ->
MAX_CONECT_SYS
#RDBCNTNUM=470          #for batch #fssqlenv -
> MAX_CONECT_SYS
RDBCNTNUM=475          #for batch #fssqlenv ->
MAX_CONECT_SYS
#NG-RDBCNTNUM=505 NG
RDBPRJCODE=0xdb

RDBSYSBUF=/rdbptc/tpcc80/SYS
RDBSQLENV=/rdbptc/tpcc80/SYS/fssqlenv
RDBLOGMANAGE=/rdbptc/tpcc80/SYS
RDBPOOLCFG=/rdbptc/tpcc80/SYS

#-------------------------------------------------------------
RDBFIXBUFMEM=80             #LOAD
#RDBFIXBUFMEMADDR=0x36000000

#old  #RDBFIXBUFMEM=2272           #sokutei
old

#2508WH RDBFIXBUFMEM=2288
#sokutei old
#RDBFIXBUFMEM=2256           # sokutei
(2772WH 990604)

RDBFIXBUFMEMADDR=0x5e000000  #
#2508WHnotoki
#RDBFIXBUFMEMADDR=0x5c000000  #
#old  #RDBFIXBUFMEMADDR=0x5d000000  #old
#-------------------------------------------------------
RDBEXTMEMADDR=0xed000000
RDBEXTMEM=8192

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

RDBKCHKSKIPCNT=200    #        IO
96.10.05
#RDBKCHKSKIPCNT=300    #        IO
96.10.05
RDBKTAJUUDOSDP = 2016
RDBKLISTNUMSDP = 5
RDBKCATENUMSDP=252
#RDBKAIOREP = 6
RDBKAIOREP = 12  #980531
#RDBKAIOREP = 18  #980531
##RDBKAIOSAV = 30
RDBKAIOYLD = 0

RDBKAIOCNT    = yes
RDBKAIOD9F    = yes
```

```
RDBKAIOSELFWAIT    = yes
RDBKAIODSP    = yes
#RDBIOERRDOWN = yes
########################################
################################nrk
##RDBVER:
UXP/DS_RDBII_V20L21_3/22_version
########################################
################################
#
RDBSDPCPU =  0,1,2,3,4,5,6,7   ## SDP
CPU
#RDBSDPCPU =  0,1,2,3,4,6   ## SDP
CPU
#RDBSDPCPU =  0,2,4,6   ## SDP              CPU
#---RDBSDPCPU =  0,1,2,3,4,6,7   ## SDP
CPU

RDBCCRDMCPU=5               ##       daemon
CPU
RDBRECEPCPU=5
#---RDBSORTCPU=5
RDBTCPIPCPU=5
RDBALFCPU=5
#RDBDBSCPU=1,2,3,4,5,6,7         ##
RDBDBSCPU=1         ##
RDBDIRCPU=5
#---RDBIOCPU=5
RDBTLFCPU=5
#---RDBWKSCPU=5



####I/O
RDBREADUNC = NO           # SAP
6/28

RDBDBSNUM+ = 350          #      /

RDBMAXLWP =  100        #98.11.11 oza
#RDBMAXDBIO = 20          # 20
I/O
RDBMAXDBIO = 10          # 990309 oza
RDBMAXRCPIO = 20          # old=20 RCP
I/O
RDBNEWPAGE = 1           # LRU
#RDBNEWPAGE = 2           # LRU
RDBANTIQUEPAGE = 4 # LRU  4             ,
RDBIOUNITNUM = 1,1
RDBSORTUNITNUM = 1,1

##LOG                    ( GC    =6     =30ms
>2 BUF=32 )
#RDBLOGAIONUM=32                #IO buffer
RDBLOGAIONUM=90                       #IO
buffer
RDBLOGBIONUM=90                 #
#RDBLOGBIONUM=256                      #
RDBLOGIOSLEEP=10                #SLTRNUM
tran  write            sleep
RDBLOGSLTRNUM=2                       #
tran         write
RDBLOGGRCOMMIT=4
        #SLEEP                 tran           nosleep

#RDBKTAJUUDOSDP=5
#RDBKCATENUMSDP=5
#RDBKCATENUMSDP=60
```

```
##SLK/LWP          tuning (0418           )
RDBSLKLOOP=10       #   100steps
#RDBLWPLOOP=100                  #   10steps
MIPS=10 (1min.=600Msteps)
RDBSEMMODE=IPC   #mutex      IPC

##my_mutex      flag
#RDBDBGSLKCNT=yes,yes
#RDBDBGSLKCNT=yes
#RDBDBGMUTCNT=yes
#RDBDBGWPCCNT = yes
#RDBDBGWAITPOS = yes
#RDBKAIOCNT    = yes

##mutex                 BUSY
#RDBMUTLOOP=100

##
mutex
.
#RDBMUTGIVE=YES
RDBIXSECDWD=YES     #

RDBSDPLDBALMODE = 2
```

File: rdbsysparm

```
#
# All Rights Reserved, Copyright(c) FUJITSU
1996
# All Rights Reserved, Copyright(c) PFU     1996
#
# Title: RDB system definition file
#
########################################
################################
# DO NOT TOUCH ME!!
#
#RDBMEMBLKSIZE=32
#RDBMEMBLKSIZE=64
RDBMEMBLKSIZE=128
RDBLBUFSIZE=0,128,512
COMMUNICATION_BUFFER=1
SORT_MEM_SIZE=64
WORK_MEM_SIZE=64
CGP_INIT_SIZE=1
CGP_ELEM=10
MEM_CMD_POOL_SIZE=1
MEM_LC1_POOL_SIZE=1
MEM_LC2_POOL_SIZE=1
MEM_LC3_POOL_SIZE=1
MEM_OPL_POOL_SIZE=1
MEM_OPT_POOL_SIZE=1
MEM_SCT_POOL_SIZE=1
MEM_SPL_POOL_SIZE=1
DYN_SQL_BUFFER=3, 1, 3
TID_BUFFER=1, 1, 3
CURSOR_NAME_BUFFER=1, 1, 1
BUFFER_SIZE=1, 1
RESULT_BUFFER=0, 1
OPL_BUFFER_SIZE=1
MAX_CONNECT_SYS=20
DESC_NUM=256
```

File: sh.crbuf

```
# 1680WH
xxxMB################################
# STOCK(781.2M->669MB)----------------------------
----
rdbcrbf -A 11250  -S 11150 -x -l 1 -m 3 S_1     4K
11650
rdbcrbf -A 11250  -S 11150 -x -l 1 -m 3 S_2     4K
11650
rdbcrbf -A 11250  -S 11150 -x -l 1 -m 3 S_3     4K
11650
rdbcrbf -A  8350  -S  8250 -x -l 1 -m 3 S_4     4K
8750
rdbcrbf -A  8350  -S  8250 -x -l 1 -m 3 S_5     4K
8750
rdbcrbf -A 11250  -S 11150 -x -l 1 -m 3 S_6     4K
11650
rdbcrbf -A 11250  -S 11150 -x -l 1 -m 3 S_7     4K
11650
rdbcrbf -A  8350  -S  8250 -x -l 1 -m 3 S_8     4K
8750
rdbcrbf -A  8350  -S  8250 -x -l 1 -m 3 S_9     4K
8750
rdbcrbf -A 11250  -S 11150 -x -l 1 -m 3 S_10    4K
11650
rdbcrbf -A  8350  -S  8250 -x -l 1 -m 3 S_11    4K
8750
rdbcrbf -A  8350  -S  8250 -x -l 1 -m 3 S_12    4K
8750
# CUSTOMER----------------------------------------------
rdbcrbf -A 1500  -S 1400 -x -l 2 -m 4 C_1     4K
1550
rdbcrbf -A 1500  -S 1400 -x -l 2 -m 4 C_2     4K
1550
rdbcrbf -A 1120  -S 1040 -x -l 2 -m 4 C_3     4K
1200
rdbcrbf -A 1120  -S 1040 -x -l 2 -m 4 C_4     4K
1200
rdbcrbf -A 1240  -S 1160 -x -l 2 -m 4 C_5     4K
1320
rdbcrbf -A 1240  -S 1160 -x -l 2 -m 4 C_6     4K
1320
rdbcrbf -A 1120  -S 1040 -x -l 2 -m 4 C_7     4K
1200
rdbcrbf -A 1120  -S 1040 -x -l 2 -m 4 C_8     4K
1200
rdbcrbf -A 1120  -S 1040 -x -l 2 -m 4 C_9     4K
1200
rdbcrbf -A 1120  -S 1040 -x -l 2 -m 4 C_10    4K
1200
rdbcrbf -A  980  -S  860 -x -l 2 -m 4 C_11    4K
1050
rdbcrbf -A  980  -S  860 -x -l 2 -m 4 C_12    4K
1050
# CUSTOMER_IX-------------------------------------------
--
rdbcrbf -A  940  -S  900 -x -l 1 -m 5 C_IX_1  8K
980
rdbcrbf -A  940  -S  900 -x -l 1 -m 5 C_IX_2  8K
980
rdbcrbf -A  700  -S  650 -x -l 1 -m 5 C_IX_3  8K
750
rdbcrbf -A  700  -S  650 -x -l 1 -m 5 C_IX_4  8K
750
rdbcrbf -A  770  -S  720 -x -l 1 -m 5 C_IX_5  8K
820
```

```
rdbcrbf -A  770  -S  720 -x -l 1 -m 5 C_IX_6  8K
820
rdbcrbf -A  700  -S  650 -x -l 1 -m 5 C_IX_7  8K
750
rdbcrbf -A  700  -S  650 -x -l 1 -m 5 C_IX_8  8K
750
rdbcrbf -A  700  -S  650 -x -l 1 -m 5 C_IX_9  8K
750
rdbcrbf -A  700  -S  650 -x -l 1 -m 5 C_IX_10 8K
750
rdbcrbf -A  620  -S  570 -x -l 1 -m 5 C_IX_11 8K
670
rdbcrbf -A  620  -S  570 -x -l 1 -m 5 C_IX_12 8K
670
# ORDERS-------------------------------------------------
rdbcrbf -A  600  -S  580 -x -l 1 -m 5 O_1     8K 620
rdbcrbf -A  600  -S  580 -x -l 1 -m 5 O_2     8K 620
rdbcrbf -A  455  -S  430 -x -l 1 -m 5 O_3     8K 480
rdbcrbf -A  455  -S  430 -x -l 1 -m 5 O_4     8K 480
rdbcrbf -A  510  -S  490 -x -l 1 -m 5 O_5     8K 530
rdbcrbf -A  510  -S  490 -x -l 1 -m 5 O_6     8K 530
rdbcrbf -A  455  -S  430 -x -l 1 -m 5 O_7     8K 480
rdbcrbf -A  455  -S  430 -x -l 1 -m 5 O_8     8K 480
rdbcrbf -A  455  -S  430 -x -l 1 -m 5 O_9     8K 480
rdbcrbf -A  455  -S  430 -x -l 1 -m 5 O_10    8K
480
rdbcrbf -A  410  -S  390 -x -l 1 -m 5 O_11    8K
430
rdbcrbf -A  410  -S  390 -x -l 1 -m 5 O_12    8K
430
# ORDER_IX-----------------------------------------------
rdbcrbf -A 1910  -S 1860 -x -l 1 -m 5 O_IX_1  4K
1960
rdbcrbf -A 1910  -S 1860 -x -l 1 -m 5 O_IX_2  4K
1960
rdbcrbf -A 1450  -S 1400 -x -l 1 -m 5 O_IX_3  4K
1500
rdbcrbf -A 1450  -S 1400 -x -l 1 -m 5 O_IX_4  4K
1500
rdbcrbf -A 1600  -S 1550 -x -l 1 -m 5 O_IX_5  4K
1650
rdbcrbf -A 1600  -S 1550 -x -l 1 -m 5 O_IX_6  4K
1650
rdbcrbf -A 1450  -S 1400 -x -l 1 -m 5 O_IX_7  4K
1500
rdbcrbf -A 1450  -S 1400 -x -l 1 -m 5 O_IX_8  4K
1500
rdbcrbf -A 1450  -S 1400 -x -l 1 -m 5 O_IX_9  4K
1500
rdbcrbf -A 1450  -S 1400 -x -l 1 -m 5 O_IX_10 4K
1500
rdbcrbf -A 1310  -S 1270 -x -l 1 -m 5 O_IX_11 4K
1350
rdbcrbf -A 1310  -S 1270 -x -l 1 -m 5 O_IX_12 4K
1350
#ORDERLINE(468M)----------------------------------------
--
rdbcrbf -A 2130  -S 2110 -x -l 1 -m 5 OL_1  32K
2150
rdbcrbf -A 2130  -S 2110 -x -l 1 -m 5 OL_2  32K
2150
rdbcrbf -A 2130  -S 2110 -x -l 1 -m 5 OL_3  32K
2150
rdbcrbf -A 2130  -S 2110 -x -l 1 -m 5 OL_4  32K
2150
rdbcrbf -A 2130  -S 2110 -x -l 1 -m 5 OL_5  32K
2150
rdbcrbf -A 2130  -S 2110 -x -l 1 -m 5 OL_6  32K
2150
rdbcrbf -A 2130  -S 2110 -x -l 1 -m 5 OL_7  32K
2150
rdbcrbf -A 2130  -S 2110 -x -l 1 -m 5 OL_8  32K
2150
rdbcrbf -A 2130  -S 2110 -x -l 1 -m 5 OL_9  32K
2150
rdbcrbf -A 2130  -S 2110 -x -l 1 -m 5 OL_10 32K
2150
rdbcrbf -A 2130  -S 2110 -x -l 1 -m 5 OL_11 32K
2150
rdbcrbf -A 2130  -S 2110 -x -l 1 -m 5 OL_12 32K
2150
#ORDERLINE(198M)----------------------------------------
rdbcrbf -A   83  -S   80 -x -l 1 -m 5 OL_1  8K 86
rdbcrbf -A   83  -S   80 -x -l 1 -m 5 OL_2  8K 86
rdbcrbf -A   83  -S   80 -x -l 1 -m 5 OL_3  8K 86
rdbcrbf -A   83  -S   80 -x -l 1 -m 5 OL_4  8K 86
rdbcrbf -A   83  -S   80 -x -l 1 -m 5 OL_5  8K 86
rdbcrbf -A   83  -S   80 -x -l 1 -m 5 OL_6  8K 86
rdbcrbf -A   83  -S   80 -x -l 1 -m 5 OL_7  8K 86
rdbcrbf -A   83  -S   80 -x -l 1 -m 5 OL_8  8K 86
rdbcrbf -A   83  -S   80 -x -l 1 -m 5 OL_9  8K 86
rdbcrbf -A   83  -S   80 -x -l 1 -m 5 OL_10 8K 86
rdbcrbf -A   83  -S   80 -x -l 1 -m 5 OL_11 8K 86
rdbcrbf -A   83  -S   80 -x -l 1 -m 5 OL_12 8K 86
#NEWORDER(128M)-----------------------------------------
--
rdbcrbf -A 2050 -S  2000 -x -l 1 -m 5 NO_1  8K
2150
rdbcrbf -A 2050 -S  2000 -x -l 1 -m 5 NO_2  8K
2150
rdbcrbf -A 2050 -S  2000 -x -l 1 -m 5 NO_3  8K
2150
rdbcrbf -A 2050 -S  2000 -x -l 1 -m 5 NO_4  8K
2150
rdbcrbf -A 2050 -S  2000 -x -l 1 -m 5 NO_5  8K
2150
rdbcrbf -A 2050 -S  2000 -x -l 1 -m 5 NO_6  8K
2150
rdbcrbf -A 2050 -S  2000 -x -l 1 -m 5 NO_7  8K
2150
rdbcrbf -A 2050 -S  2000 -x -l 1 -m 5 NO_8  8K
2150
rdbcrbf -A 2050 -S  2000 -x -l 1 -m 5 NO_9  8K
2150
rdbcrbf -A 2050 -S  2000 -x -l 1 -m 5 NO_10 8K
2150
rdbcrbf -A 2050 -S  2000 -x -l 1 -m 5 NO_11 8K
2150
rdbcrbf -A 2050 -S  2000 -x -l 1 -m 5 NO_12 8K
2150
#NEWORDER_IX(166MB)-------------------------------------
---
rdbcrbf -A 3150 -S  3100 -x -l 1 -m 5 NO_IX_1  8K
3250
rdbcrbf -A 3150 -S  3100 -x -l 1 -m 5 NO_IX_2  8K
3250
rdbcrbf -A 3150 -S  3100 -x -l 1 -m 5 NO_IX_3  8K
3250
rdbcrbf -A 3150 -S  3100 -x -l 1 -m 5 NO_IX_4  8K
3250
rdbcrbf -A 3150 -S  3100 -x -l 1 -m 5 NO_IX_5  8K
3250
rdbcrbf -A 3150 -S  3100 -x -l 1 -m 5 NO_IX_6  8K
3250
rdbcrbf -A 3150 -S  3100 -x -l 1 -m 5 NO_IX_7  8K
3250
rdbcrbf -A 3150 -S  3100 -x -l 1 -m 5 NO_IX_8  8K
3250
rdbcrbf -A 3150 -S  3100 -x -l 1 -m 5 NO_IX_9  8K
3250
rdbcrbf -A 3150 -S  3100 -x -l 1 -m 5 NO_IX_10 8K
3250
rdbcrbf -A 3150 -S  3100 -x -l 1 -m 5 NO_IX_11 8K
3250
rdbcrbf -A 3150 -S  3100 -x -l 1 -m 5 NO_IX_12 8K
3250
#HISTORY(3M)--------------------------------------
rdbcrbf -A 260 -S 250 -x -l 1 -m 5 H_1  4K 280
rdbcrbf -A 260 -S 250 -x -l 1 -m 5 H_2  4K 280
rdbcrbf -A 190 -S 180 -x -l 1 -m 5 H_3  4K 210
rdbcrbf -A 190 -S 180 -x -l 1 -m 5 H_4  4K 210
rdbcrbf -A 190 -S 180 -x -l 1 -m 5 H_5  4K 210
rdbcrbf -A 190 -S 180 -x -l 1 -m 5 H_6  4K 210
rdbcrbf -A 190 -S 180 -x -l 1 -m 5 H_7  4K 210
rdbcrbf -A 190 -S 180 -x -l 1 -m 5 H_8  4K 210
rdbcrbf -A 190 -S 180 -x -l 1 -m 5 H_9  4K 210
rdbcrbf -A 190 -S 180 -x -l 1 -m 5 H_10 4K 210
rdbcrbf -A 190 -S 180 -x -l 1 -m 5 H_11 4K 210
rdbcrbf -A 190 -S 180 -x -l 1 -m 5 H_12 4K 210
```

File: sh.crbuf.mbs

```
rm mbs_tmp/*_*
echo "0 37100">mbs_tmp/S_12
echo "1 37100">mbs_tmp/S_22
echo "2 37100">mbs_tmp/S_32
echo "2 27800">mbs_tmp/S_42
echo "3 27800">mbs_tmp/S_52
echo "3 37100">mbs_tmp/S_62
echo "4 37100">mbs_tmp/S_72
echo "4 27800">mbs_tmp/S_82
echo "5 27800">mbs_tmp/S_92
echo "5 37100">mbs_tmp/S_102
echo "6 27800">mbs_tmp/S_112
echo "6 27800">mbs_tmp/S_122
```

File: sh.rdbrcp

```
#!/bin/sh
#
# rdbrcp            $1=     $2=     $3=No.
rscnt=1
while [ $rscnt -le $2 ]
do
        sleep $1
        sh.rdbrcp.sub $3 &
        rscnt=`expr $rscnt + 1`
done
wait
```

File: sh.rdbrcp.sub

```
#!/bin/sh
#
# rdbrcp        $1=No.
        echo 'RCP-START---'   >>res.doc/ckpt.$1
```

```
date            >>res.doc/ckpt.$1
date
echo 'RCP-START----------------------------------
-------------'
timex rdbrcp
echo 'RCP-END--------------------------------------
-----------'
date
date            >>res.doc/ckpt.$1
echo 'RCP-END---'    >>res.doc/ckpt.$1
```

File: sql.env

```
;
; All Rights Reserved, Copyright(c) FUJITSU 1993,
1994, 1995
; All Rights Reserved, Copyright(c) PFU    1993,
1994, 1995
;
;    :
;
;
;          :        ';'
;          :
;          : 1                    1024
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;     UXPSQLENV
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SERVER_SPEC = ( RDB2_TCP, SV1, TPCC ,
pcrdbsv10 , 2050 )
DEFAULT_CONNECTION = ( TPCC,
Administrator, rdb2 )

TRAN_SPEC = ( TRANSACTION_ROLLBACK )
;BUFFER_SIZE = ( 128 , 128 )
;WAIT_TIME = ( 0 )
;PERFORMANCE = (OFF,
/SORTWK2/temp.perf )
;NCHAR_CODE = ( EUC )
;OPL_BUFFER_SIZE = ( 256 )
OPL_BUFFER_SIZE = ( 280 )
;CHARACTER_TRANSLATE = CLIENT
;CHAR_CODE = EUC
;RESULT_BUFFER = ( 5, 16 )
RESULT_BUFFER = ( 0 )
;MSG_PRINT = ( ON )
;DSO_LOCK =
( TPCC.DISTRICT_DSO/PEX,TPCC.STOCK_DS
O/PEX,TPCC.ORDERLINE_IX_DSO/PEX,TPCC.
ORDERS_DSO/PEX,TPCC.ORDERS_IX1_DSO/
PEX,TPCC.ORDERS_IX2_DSO/PEX,TPCC.ORD
ERLINE_DSO/PEX,TPCC.CUSTOMER_IX_DSO/
SH,TPCC.ITEM_DSO/SH )
;DSO_LOCK =
( TPCC.HISTORY_DSO/EX,TPCC.CUSTOMER_I
X_DSO/SH,TPCC.ITEM_DSO/SH,TPCC.ORDER
LINE_DSO/EX,TPCC.ORDERLINE_IX_DSO/EX )
;DSO_LOCK =
( TPCC.HISTORY_DSO/EX,TPCC.CUSTOMER_I
X_DSO/SH,TPCC.ITEM_DSO/SH )
;DSO_LOCK =
( TPCC.HISTORY_DSO/EX,TPCC.CUSTOMER_I
X_DSO/SH,TPCC.ITEM_DSO/SH,
;
    TPCC.ORDERS_DSO/EX,TPCC.ORDERS_
IX_DSO/EX )
;
;DSO_LOCK =
( TPCC.HISTORY_DSO/EX,TPCC.CUSTOMER_I
X_DSO/SH,TPCC.ITEM_DSO/SH )
;
DSO_LOCK =
( TPCC.HISTORY_DSO/EX,TPCC.CUSTOMER_I
X_DSO/SH,TPCC.ITEM_DSO/SH,
    TPCC.ORDERLINE_DSO/EX,TPCC.ORDE
RS_DSO/EX,TPCC.ORDERS_IX_DSO/EX,
    TPCC.NEWORDER_DSO/EX,TPCC.NEWO
RDER_IX_DSO/EX )
SIGNAL_INF = NO
SORT_MEM_SIZE = 128
WORK_MEM_SIZE = 64

;ROUTINE_SNAP = (ON,c:\R_SNAP.txt,2)
;SQL_SNAP = (ALL, snap.txt )
;SQL_SNAP = (OFF, /SORTWK2/temp.snap )
```

File: system

```
*ident "@(#)system      1.18  97/06/27 SMI" /*
SVR4 1.5 */
*
* SYSTEM SPECIFICATION FILE
*

* moddir:
*
*     Set the search path for modules.  This has a
format similar to the
*     csh path variable. If the module isn't found
in the first directory
*     it tries the second and so on. The default is
/kernel /usr/kernel
*
*     Example:
*          moddir: /kernel /usr/kernel
/other/modules



* root device and root filesystem configuration:
*
*     The following may be used to override the
defaults provided by
*     the boot program:
*
*     rootfs:          Set the filesystem type of
the root.
*
*     rootdev:    Set the root device.  This should
be a fully
*                 expanded physical pathname.
The default is the
*                 physical pathname of the device
where the boot
*                 program resides.  The physical
pathname is
*                 highly platform and
configuration dependent.
*
*     Example:
*          rootfs:ufs
*
     rootdev:/sbus@1,f8000000/esp@0,800000/
sd@3,0:a
*
*     (Swap device configuration should be
specified in /etc/vfstab.)



* exclude:
*
*     Modules appearing in the moddir path which
are NOT to be loaded,
*     even if referenced. Note that `exclude'
accepts either a module name,
*     or a filename which includes the directory.
*
*     Examples:
*          exclude: win
*          exclude: sys/shmsys



* forceload:
*
*     Cause these modules to be loaded at boot
time, (just before mounting
*     the root filesystem) rather than at first
reference. Note that
*     forceload expects a filename which includes
the directory. Also
*     note that loading a module does not
necessarily imply that it will
*     be installed.
*
*     Example:
*          forceload: drv/foo



* set:
*
*     Set an integer variable in the kernel or a
module to a new value.
*     This facility should be used with caution.
See system(4).
*
*     Examples:
*
*     To set variables in 'unix':
*
*          set nautopush=32
*          set maxusers=40
*
*     To set a variable named 'debug' in the
module named 'test_module'
*
*          set test_module:debug = 0x13

set msgsys:msginfo_msgmap = 200
set msgsys:msginfo_msgmax = 16384
set msgsys:msginfo_msgmnb = 32768
set msgsys:msginfo_msgmni = 512
set msgsys:msginfo_msgseg = 31744

set semsys:seminfo_semmni = 24576
```

```
set semsys:seminfo_semmns = 18432
set semsys:seminfo_semmnu = 6144
set semsys:seminfo_semmsl = 32

set shmsys:shminfo_shmmax = 2097152000
set shmsys:shminfo_shmmni = 1024
set shmsys:shminfo_shmseg = 512

* TCP-IP Max User
set pt_cnt = 300

set  enable_grp_ism = 1
set dump_userproc = 0


*PA
forceload: sys/perfmon
```

File: tra.wup.pc

```
/* TPCC                         96.06.18     */
/*    :                              . */
#define DEC_TO_INT   /*       DEC  INT
*/

EXEC SQL BEGIN DECLARE SECTION;
char          SQLSTATE[6];
int i_c;
short w_id;
short we_id;
short d_id;
int   o_id;
int item_s;
int item_e;
EXEC SQL END DECLARE SECTION;

main( int argc, char *argv[] ){
        int i, j;
        int ws=1,we=80;
        char sqlnormal[6] = "00000";
        if( argc == 3 ){
                ws = atoi( argv[1]);
                we = atoi( argv[2]);
        }
        system("date");

        if( ws == 0 ){
                if( we != 0 ){
                        printf("  (1)
(  25          )\n");
                        item_s = we * 10000 - 10000 +
1 ;
                        item_e = we * 10000;
                        printf("I_ID=%d  %d
\n",item_s,item_e);
                        EXEC SQL SELECT COUNT(*)
INTO :i_c  FROM TPCC_SCHEMA.ITEM
                                        WHERE  I_ID
        > :item_s AND I_ID < :item_e;
                } else {
                        printf("  (1)
(  90          )\n");
                        EXEC SQL SELECT COUNT(*)
INTO :i_c  FROM TPCC_SCHEMA.ITEM;
                }
                EXEC SQL COMMIT WORK ;
                        exit(0);
        }

}
```

File: ubbconfig

```
#
#       ubbconfig : TUXEDO configuration file
#

*RESOURCES
IPCKEY          133133
MASTER          SITE1
#UID        1
#GID        1
PERM            0660
MAXACCESSERS        800
MAXSERVERS      92
MAXSERVICES     1000
MODEL           SHM
LDBAL           Y

*MACHINES
NTCL14      LMID=SITE1
        APPDIR="c:\sv-apl\fml"
        TUXCONFIG="c:\client\tuxconfig"
        TUXDIR="c:\tuxedo"
        ULOGPFX="c:\tuxlog\numazu"
        ENVFILE="c:\sv-apl\fml\envfile.txt"

*GROUPS
group1      LMID=SITE1 GRPNO=1
group2      LMID=SITE1 GRPNO=2
group3      LMID=SITE1 GRPNO=3
group4      LMID=SITE1 GRPNO=4
group5      LMID=SITE1 GRPNO=5
group6      LMID=SITE1 GRPNO=6
group7      LMID=SITE1 GRPNO=7
group8      LMID=SITE1 GRPNO=8
group9      LMID=SITE1 GRPNO=9
group10     LMID=SITE1 GRPNO=10
group11     LMID=SITE1 GRPNO=11
group12     LMID=SITE1 GRPNO=12
group13     LMID=SITE1 GRPNO=13
group14     LMID=SITE1 GRPNO=14
group15     LMID=SITE1 GRPNO=15
group16     LMID=SITE1 GRPNO=16
group17     LMID=SITE1 GRPNO=17
group18     LMID=SITE1 GRPNO=18
group19     LMID=SITE1 GRPNO=19
group20     LMID=SITE1 GRPNO=20
group21     LMID=SITE1 GRPNO=21
group22     LMID=SITE1 GRPNO=22
group23     LMID=SITE1 GRPNO=23
group24     LMID=SITE1 GRPNO=24
group25     LMID=SITE1 GRPNO=25
group26     LMID=SITE1 GRPNO=26
group27     LMID=SITE1 GRPNO=27
group28     LMID=SITE1 GRPNO=28
group29     LMID=SITE1 GRPNO=29
group30     LMID=SITE1 GRPNO=30
group31     LMID=SITE1 GRPNO=31
group32     LMID=SITE1 GRPNO=32
group33     LMID=SITE1 GRPNO=33
group34     LMID=SITE1 GRPNO=34
group35     LMID=SITE1 GRPNO=35
group36     LMID=SITE1 GRPNO=36
group37     LMID=SITE1 GRPNO=37
group38     LMID=SITE1 GRPNO=38
group39     LMID=SITE1 GRPNO=39
group40     LMID=SITE1 GRPNO=40
group41     LMID=SITE1 GRPNO=41
group42     LMID=SITE1 GRPNO=42
group43     LMID=SITE1 GRPNO=43
group44     LMID=SITE1 GRPNO=44
group45     LMID=SITE1 GRPNO=45
group46     LMID=SITE1 GRPNO=46

*SERVERS
DEFAULT:  RESTART=Y MAXGEN=5
REPLYQ=N RQPERM=0660
tpcc_NT_fml       SRVGRP=group1
RQADDR=TPCCq1 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group2
RQADDR=TPCCq2 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group3
RQADDR=TPCCq3 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group4
RQADDR=TPCCq4 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group5
RQADDR=TPCCq5 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group6
RQADDR=TPCCq6 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group7
RQADDR=TPCCq7 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group8
RQADDR=TPCCq8 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group9
RQADDR=TPCCq9 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group10
RQADDR=TPCCq10 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group11
RQADDR=TPCCq11 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group12
RQADDR=TPCCq12 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group13
RQADDR=TPCCq13 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group14
RQADDR=TPCCq14 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group15
RQADDR=TPCCq15 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group16
RQADDR=TPCCq16 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group17
RQADDR=TPCCq17 SRVID=1 CLOPT="-s
TPCC:TPCC"
```

```
tpcc_NT_fml       SRVGRP=group18
RQADDR=TPCCq18 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group19
RQADDR=TPCCq19 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group20
RQADDR=TPCCq20 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group21
RQADDR=TPCCq21 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group22
RQADDR=TPCCq22 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group23
RQADDR=TPCCq23 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group24
RQADDR=TPCCq24 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group25
RQADDR=TPCCq25 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group26
RQADDR=TPCCq26 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group27
RQADDR=TPCCq27 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group28
RQADDR=TPCCq28 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group29
RQADDR=TPCCq29 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group30
RQADDR=TPCCq30 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group31
RQADDR=TPCCq31 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group32
RQADDR=TPCCq32 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group33
RQADDR=TPCCq33 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group34
RQADDR=TPCCq34 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group35
RQADDR=TPCCq35 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group36
RQADDR=TPCCq36 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group37
RQADDR=TPCCq37 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group38
RQADDR=TPCCq38 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group39
RQADDR=TPCCq39 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group40
RQADDR=TPCCq40 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group41
RQADDR=TPCCq41 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group42
RQADDR=TPCCq42 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group43
RQADDR=TPCCq43 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group44
RQADDR=TPCCq44 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group45
RQADDR=TPCCq45 SRVID=1 CLOPT="-s
TPCC:TPCC"
tpcc_NT_fml       SRVGRP=group46
RQADDR=TPCCq46 SRVID=1 CLOPT="-s
TPCC:TPCC"

*SERVICES
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group1
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group2
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group3
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group4
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group5
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group6
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group7
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group8
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group9
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group10
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group11
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group12
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group13
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group14
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group15
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group16
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group17
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group18
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group19
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group20
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group21
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group22
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group23
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group24
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group25
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group26
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group27
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group28
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group29
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group30
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group31
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group32
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group33
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group34
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group35
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group36
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group37
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group38
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group39
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group40
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group41
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group42
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group43
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group44
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group45
"TPCC"      TRANTIME=0      ROUTING="route"
SRVGRP=group46

*ROUTING
"route" FIELD=FML_TERM
        BUFTYPE="FML"
        RANGES="1-11:group1,12-22:group2,23-
33:group3,34-44:group4,45-55:group5,56-
66:group6,67-77:group7,78-88:group8,89-
99:group9,100-110:group10,111-121:group11,122-
132:group12,133-143:group13,144-
154:group14,155-165:group15,166-
176:group16,177-187:group17,188-
198:group18,199-209:group19,210-
220:group20,221-231:group21,232-
242:group22,243-253:group23,254-
264:group24,265-275:group25,276-
286:group26,287-297:group27,298-
308:group28,309-319:group29,320-
330:group30,331-341:group31,342-
352:group32,353-363:group33,364-
374:group34,375-385:group35,386-
```

```
396:group36,397-407:group37,408-
418:group38,419-429:group39,430-
440:group40,441-451:group41,452-
462:group42,463-473:group43,474-
484:group44,485-495:group45,496-
506:group46,*:*"
```

File: wupi.sh

```
:
set -x
date
tra.wup 0 1 & tra.wup 0 2 & tra.wup 0 3 & tra.wup
0 4 & tra.wup 0 5 & tra.wup 0 6 & tra.wup 0 7 &
tra.wup 0 8 & tra.wup 0 9 & tra.wup 0 10 &
#tra.wup 0 1 &
wait
date
```

Appendix E: Database Creation Code

File: ALL.mk

```
:
set -x

rdbstop

CRDIC

date
rdbstart
rdbddlex ddl_db.mak
rdbddlex -d TPCC crta.def.cent

rdbddlex -d TPCC ddl.dbsp.dat

rdbddlex -d TPCC ddl.dat.all

sleep 5
rdbstop
date

rdbstart
sh.stored
sleep 5
rdbstop

rdbstart
timex csh -x LOAD.sh
rdbstop

rdbstart
sh.rdbups
rdbstop

sleep 5

sh mktmplog.sh
sh mkarc.sh
```

File: CRDIC

```
:
set -x
SYS=/rdbptc/tpcc80/SYS
LOG_AI=/DEV/rdsk/AI_LOG
LOG_BI=/DEV/rdsk/BI_LOG
LOG_IX=/DEV/rdsk/IX_LOG
DIC_PL=/DEV/rdsk/DIC

BISZ=64M
AISZ=64M
TRN=100
RCV=32M

date
rm -f $SYS/rdblogmanage

rdblog -I
timex rdblog -G -t -c $RCV -io 2048 $LOG_IX
$LOG_BI $LOG_AI $BISZ $AISZ $TRN
timex rdbcrdic -du 37M -r $DIC_PL
date

rdbstart
rdbstop
```

File: crta.def.cent

```
--
##########################################
##################################
-- #    TPC-C
-- #
-- #    1995.5.15
-- #    1996.4.18                DECIMAL -> CHAR
or SMALLINT or INTEGER
-- #    1996.10.18
       C_ID,H_C_ID,O_C_ID SMALLINT ->
INTEGER
-- #                                    I_IM_ID
-- #    1998.11.24          W_YTD, D_YTD,
C_BALANCE, C_CREDIT_LIM,
-- #                        C_YTD_PAYMENT      -
>    12
-- #
--
##########################################
##################################
CREATE SCHEMA TPCC_SCHEMA
--
##########################################
##################################
-- #
--
##########################################
##################################
CREATE TABLE
TPCC_SCHEMA.WAREHOUSE(
       W_ID         SMALLINT  NOT NULL,
       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        DECIMAL(4,4)  NOT
NULL,       '96/04/18
       W_TAX        SMALLINT  NOT NULL,
--     W_YTD        DECIMAL(12,2) NOT
NULL,       '98/11/24
       W_YTD        DECIMAL(12,0) NOT
NULL,
       PRIMARY KEY(W_ID)
)

CREATE TABLE TPCC_SCHEMA.DISTRICT(
       D_ID         SMALLINT  NOT NULL,
       D_W_ID       SMALLINT  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_TAX        DECIMAL(4,4)  NOT
NULL,       '96/04/18
       D_TAX        SMALLINT  NOT NULL,
--     D_YTD        DECIMAL(12,2) NOT
NULL,       '98/11/24
       D_YTD        DECIMAL(12,0) NOT
NULL,
       D_NEXT_O_ID  INTEGER       NOT
NULL,
       PRIMARY KEY(D_W_ID,D_ID)
)

CREATE TABLE TPCC_SCHEMA.CUSTOMER(
--     C_ID         SMALLINT  NOT NULL,
       '96/10/18
       C_ID         INTEGER       NOT NULL,
       C_D_ID       SMALLINT  NOT NULL,
       C_W_ID       SMALLINT  NOT NULL,
       C_FIRST      CHAR(16)  NOT NULL,
       C_MIDDLE CHAR(2)       NOT NULL,
       C_LAST       CHAR(16)  NOT NULL,
       C_STREET_1   CHAR(20)  NOT NULL,
       C_STREET_2   CHAR(20)  NOT NULL,
       C_CITY       CHAR(20)  NOT NULL,
       C_STATE      CHAR(2)       NOT
NULL,
       C_ZIP        CHAR(9)       NOT
NULL,
       C_PHONE      CHAR(16)  NOT NULL,
--     C_SINCE      DECIMAL(14)   NOT
NULL,          '96/04/18
       C_SINCE      CHAR(14)  NOT NULL,
       C_CREDIT CHAR(2)       NOT NULL,
--     C_CREDIT_LIM DECIMAL(12,2) NOT
NULL,       '98/11/24
       C_CREDIT_LIM DECIMAL(12,0) NOT
NULL,
--     C_DISCOUNT   DECIMAL(4,4)  NOT
NULL,       '96/04/18
       C_DISCOUNT   SMALLINT  NOT NULL,
--     C_BALANCE    DECIMAL(12,2) NOT
NULL,       '98/11/24
       C_BALANCE    DECIMAL(12,0) NOT
NULL,
--     C_YTD_PAYMENT      DECIMAL(12,2)
       NOT NULL,'98/11/24
       C_YTD_PAYMENT      DECIMAL(12,0)
       NOT NULL,
       C_PAYMENT_CNT      SMALLINT  NOT
NULL,
       C_DELIVERY_CNT     SMALLINT  NOT
NULL,
       C_DATA       CHAR(500) NOT NULL,
       PRIMARY KEY(C_W_ID, C_D_ID, C_ID)
)

CREATE TABLE TPCC_SCHEMA.ITEM(
       I_ID         INTEGER       NOT NULL,
       I_IM_ID      INTEGER       NOT
NULL,
       I_NAME       CHAR(24)  NOT NULL,
```

```
--      I_PRICE         DECIMAL(5,2)    NOT
NULL,       '96/04/18
        I_PRICE         SMALLINT  NOT NULL,
        I_DATA          CHAR(50)  NOT NULL,
        PRIMARY KEY(I_ID)
)

CREATE TABLE TPCC_SCHEMA.STOCK(
        S_I_ID          INTEGER         NOT
NULL,
        S_W_ID          SMALLINT  NOT NULL,
        S_QUANTITY      SMALLINT  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_YTD           INTEGER         NOT
NULL,
        S_ORDER_CNT  SMALLINT  NOT NULL,
        S_REMOTE_CNT        SMALLINT  NOT
NULL,
        S_DATA          CHAR(50)  NOT NULL,
        PRIMARY KEY(S_W_ID, S_I_ID)
)

CREATE TABLE TPCC_SCHEMA.NEWORDER(
        NO_O_ID         INTEGER         NOT
NULL,
        NO_D_ID         SMALLINT  NOT NULL,
        NO_W_ID         SMALLINT  NOT NULL,
        PRIMARY KEY(NO_W_ID, NO_D_ID,
NO_O_ID)
)

CREATE TABLE TPCC_SCHEMA.ORDERS(
        O_ID        INTEGER         NOT NULL,
        O_D_ID          SMALLINT  NOT NULL,
        O_W_ID          SMALLINT  NOT NULL,
--      O_C_ID          SMALLINT  NOT NULL,
        '96/10/18
        O_C_ID          INTEGER         NOT
NULL,
--      O_ENTRY_D       DECIMAL(14)     NOT
NULL,           '96/04/18
        O_ENTRY_D       CHAR(14)  NOT NULL,
        O_CARRIER_ID  SMALLINT,
        O_OL_CNT SMALLINT  NOT NULL,
        O_ALL_LOCAL     SMALLINT  NOT NULL,
        PRIMARY KEY(O_W_ID, O_D_ID, O_ID)
)

CREATE TABLE TPCC_SCHEMA.ORDERLINE(
        OL_O_ID         INTEGER         NOT
NULL,
        OL_D_ID         SMALLINT  NOT NULL,
        OL_W_ID         SMALLINT  NOT NULL,
        OL_NUMBER       SMALLINT  NOT NULL,
        OL_I_ID         INTEGER         NOT
NULL,
        OL_SUPPLY_W_ID      SMALLINT  NOT
NULL,
--      OL_DELIVERY_D       DECIMAL(14),
        '96/04/18
        OL_DELIVERY_D       CHAR(14),
        OL_QUANTITY     SMALLINT  NOT NULL,
--      OL_AMOUNT       DECIMAL(6,2)    NOT
NULL,       '96/04/18
        OL_AMOUNT       INTEGER         NOT
NULL,
        OL_DIST_INFO    CHAR(24)  NOT NULL,
--      PRIMARY KEY(OL_W_ID, OL_D_ID,
OL_O_ID, OL_NUMBER)
        PRIMARY KEY(OL_W_ID, OL_D_ID,
OL_NUMBER, OL_O_ID )
)

CREATE TABLE TPCC_SCHEMA.HISTORY(
--      H_C_ID          SMALLINT  NOT NULL,
        '96/10/18
        H_C_ID          INTEGER         NOT
NULL,
        H_C_D_ID  SMALLINT  NOT NULL,
        H_C_W_ID SMALLINT  NOT NULL,
        H_D_ID          SMALLINT  NOT NULL,
        H_W_ID          SMALLINT  NOT NULL,
--      H_DATE          DECIMAL(14)     NOT
NULL,           '96/04/27
        H_DATE          CHAR(14)  NOT NULL,
--      H_AMOUNT        DECIMAL(6,2)    NOT
NULL,       '96/04/18
        H_AMOUNT        INTEGER   NOT NULL,
        H_DATA          CHAR(24)  NOT NULL
)
```

File: ddl.dat.all

```
-------------------------------------------------------------------
---------
-- * Phase.2: DSO/DSI definition
-------------------------------------------------------------------
---------
-------------------------------------------------------------------
---------
-- * Phase.2-1: Warehouse
-------------------------------------------------------------------
---------
        CREATE DSO WAREHOUSE_DSO
                FROM
TPCC_SCHEMA.WAREHOUSE
                        TYPE
RANDOM(PAGESIZE1(1),PAGESIZE2(1))
                        WHERE (W_ID) BETWEEN (?)
AND (?);

        CREATE DSI WAREHOUSE_1_DSI
            DSO WAREHOUSE_DSO
              USING(1,66)
              ALLOCATE PRIME    ON SP201
SIZE 2049K,
                        OVERFLOW ON SP201 SIZE
63K;

        CREATE DSI WAREHOUSE_2_DSI
            DSO WAREHOUSE_DSO
              USING(67,132)
              ALLOCATE PRIME    ON SP202
SIZE 2049K,
                        OVERFLOW ON SP202 SIZE
63K;

    .
    .
    .

        CREATE DSI WAREHOUSE_41_DSI
            DSO WAREHOUSE_DSO
              USING(2641,2706)
              ALLOCATE PRIME    ON SP241
SIZE 2049K,
                        OVERFLOW ON SP241 SIZE
63K;

        CREATE DSI WAREHOUSE_42_DSI
            DSO WAREHOUSE_DSO
              USING(2707,5544)
              ALLOCATE PRIME    ON SP242
SIZE 2049K,
                        OVERFLOW ON SP242 SIZE
63K;

-------------------------------------------------------------------
---------
-- * Phase.2-2: District
-------------------------------------------------------------------
---------
        CREATE DSO DISTRICT_DSO
                FROM TPCC_SCHEMA.DISTRICT
                        TYPE
RANDOM(PAGESIZE1(1),PAGESIZE2(1),RULE(
D_W_ID*20+D_ID*2))
                        WHERE (D_W_ID) BETWEEN
(?) AND (?);

        CREATE DSI DISTRICT_1_DSI
            DSO DISTRICT_DSO
              USING(1,66)
              ALLOCATE PRIME    ON SP201
SIZE 1321K,
                        OVERFLOW ON SP201 SIZE
23K;

        CREATE DSI DISTRICT_2_DSI
            DSO DISTRICT_DSO
              USING(67,132)
              ALLOCATE PRIME    ON SP202
SIZE 1321K,
                        OVERFLOW ON SP202 SIZE
23K;

    .
    .
    .

        CREATE DSI DISTRICT_41_DSI
            DSO DISTRICT_DSO
              USING(2641,2706)
              ALLOCATE PRIME    ON SP241
SIZE 1321K,
                        OVERFLOW ON SP241 SIZE
23K;

        CREATE DSI DISTRICT_42_DSI
            DSO DISTRICT_DSO
              USING(2707,5544)
```

```
            ALLOCATE PRIME    ON SP242
SIZE 1321K,
                OVERFLOW ON SP242 SIZE
23K;

-------------------------------------------------------------------
---------
-- * Phase.2-8: Stock
-------------------------------------------------------------------
---------
    CREATE DSO STOCK_DSO
            FROM TPCC_SCHEMA.STOCK
                TYPE
RANDOM(PAGESIZE1(4),PAGESIZE2(1),
                RULE(S_I_ID*3+(S_W_ID-
1)/11+(S_W_ID-S_W_ID/11*11)*300000))
                WHERE (S_W_ID) BETWEEN
(?) AND (?);

    CREATE DSI STOCK_1_DSI
        DSO STOCK_DSO
          USING(1,33)
          ALLOCATE PRIME    ON SP1 SIZE
400004K
                                    SP2
SIZE 400000K
                                    SP3
SIZE 400000K,
                OVERFLOW ON SP1 SIZE
60028K;

    CREATE DSI STOCK_2_DSI
        DSO STOCK_DSO
          USING(34,66)
          ALLOCATE PRIME    ON SP1 SIZE
400000K
                                    SP2
SIZE 400004K
                                    SP3
SIZE 400000K,
                OVERFLOW ON SP2 SIZE
60028K;

    CREATE DSI STOCK_3_DSI
        DSO STOCK_DSO
          USING(67,99)
          ALLOCATE PRIME    ON SP4 SIZE
400004K
                                    SP5
SIZE 400000K
                                    SP6
SIZE 400000K,
                OVERFLOW ON SP4 SIZE
60028K;

    CREATE DSI STOCK_4_DSI
        DSO STOCK_DSO
          USING(100,132)
          ALLOCATE PRIME    ON SP4 SIZE
400000K
                                    SP5
SIZE 400004K
                                    SP6
SIZE 400000K,
                OVERFLOW ON SP5 SIZE
60028K;

  .
  .
  .

    CREATE DSI STOCK_81_DSI
        DSO STOCK_DSO
          USING(2641,2673)
          ALLOCATE PRIME    ON SP121
SIZE 400004K
                                    SP122
SIZE 400000K
                                    SP123
SIZE 400000K,
                OVERFLOW ON SP121 SIZE
60028K;

    CREATE DSI STOCK_82_DSI
        DSO STOCK_DSO
          USING(2674,2706)
          ALLOCATE PRIME    ON SP121
SIZE 400000K
                                    SP122
SIZE 400004K
                                    SP123
SIZE 400000K,
                OVERFLOW ON SP122 SIZE
60028K;

    CREATE DSI STOCK_83_DSI
        DSO STOCK_DSO
          USING(2707,2739)
          ALLOCATE PRIME    ON SP124
SIZE 400004K
                                    SP125
SIZE 400000K
                                    SP126
SIZE 400000K,
                OVERFLOW ON SP124 SIZE
60028K;

    CREATE DSI STOCK_84_DSI
        DSO STOCK_DSO
          USING(2740,5544)
          ALLOCATE PRIME    ON SP124
SIZE 400000K
                                    SP125
SIZE 400004K
                                    SP126
SIZE 400000K,
                OVERFLOW ON SP125 SIZE
60028K;

-------------------------------------------------------------------
---------
-- * Phase.2-6a: NewOrder
-------------------------------------------------------------------
---------
    CREATE DSO NEWORDER_DSO
            FROM TPCC_SCHEMA.NEWORDER
          TYPE
RANDOM(PAGESIZE1(8),PAGESIZE2(1),RULE((
NO_O_ID/8)*11+NO_W_ID+((NO_D_ID-
1)*8+(NO_O_ID-((NO_O_ID/8)*8)))*1799))
                WHERE (NO_W_ID)
BETWEEN (?) AND (?);

    CREATE DSI NEWORDER_1_DSI
        DSO NEWORDER_DSO
          USING(1,11)
            ALLOCATE PRIME    ON SP201
SIZE 14400K,
                OVERFLOW ON SP201 SIZE
448K;

    CREATE DSI NEWORDER_2_DSI
        DSO NEWORDER_DSO
          USING(12,22)
          ALLOCATE PRIME    ON SP202
SIZE 14400K,
                OVERFLOW ON SP202 SIZE
448K;

  .
  .
  .

    CREATE DSI NEWORDER_251_DSI
        DSO NEWORDER_DSO
          USING(2751,2761)
          ALLOCATE PRIME    ON SP263
SIZE 14400K,
                OVERFLOW ON SP263 SIZE
448K;

    CREATE DSI NEWORDER_252_DSI
        DSO NEWORDER_DSO
          USING(2762,5544)
          ALLOCATE PRIME    ON SP264
SIZE 14400K,
                OVERFLOW ON SP264 SIZE
448K;

-------------------------------------------------------------------
---------
-- * Phase.2-6b: NewOrder-Index
-------------------------------------------------------------------
---------
    CREATE DSO NEWORDER_IX_DSO
            INDEX ON
TPCC_SCHEMA.NEWORDER(NO_W_ID,NO_D_
ID,NO_O_ID)
            TYPE
BTREE(PAGESIZE1(8),PAGESIZE2(32),DEGENE
RATE);

    CREATE DSI NEWORDER_X_1_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_1_DSI
            ALLOCATE INDEX    ON SP201 SIZE
128K,
                BASE    ON SP201 SIZE
6656K;

    CREATE DSI NEWORDER_X_2_DSI
            INDEX
            DSO NEWORDER_IX_DSO
            BASE NEWORDER_2_DSI
            ALLOCATE INDEX    ON SP202 SIZE
128K,
                BASE    ON SP202 SIZE
6656K;

  .
  .
  .
```

```
        CREATE DSI NEWORDER_X_251_DSI
                INDEX
                DSO NEWORDER_IX_DSO
                BASE NEWORDER_251_DSI
                ALLOCATE INDEX    ON SP263 SIZE
128K,
                        BASE     ON SP263 SIZE
6656K;

        CREATE DSI NEWORDER_X_252_DSI
                INDEX
                DSO NEWORDER_IX_DSO
                BASE NEWORDER_252_DSI
                ALLOCATE INDEX    ON SP264 SIZE
128K,
                        BASE     ON SP264 SIZE
6656K;

-------------------------------------------------------------------
---------
-- * Phase.2-4a: Orders
-------------------------------------------------------------------
---------
        CREATE DSO ORDERS_DSO
                FROM TPCC_SCHEMA.ORDERS
                        TYPE
RANDOM(PAGESIZE1(8),PAGESIZE2(1),RULE((
O_ID/8)*11+O_W_ID+((O_D_ID-1)*10+(O_ID-
((O_ID/8)*8)))*5247))
                        WHERE (O_W_ID) BETWEEN
(?) AND (?);

        CREATE DSI ORDERS_1_DSI
                DSO ORDERS_DSO
                  USING(1,11)
                  ALLOCATE PRIME    ON SP1 SIZE
41984K,
                        OVERFLOW ON SP1 SIZE
320K;

        CREATE DSI ORDERS_2_DSI
                DSO ORDERS_DSO
                  USING(12,22)
                  ALLOCATE PRIME    ON SP1 SIZE
41984K,
                        OVERFLOW ON SP1 SIZE
320K;

    .
    .
    .

        CREATE DSI ORDERS_251_DSI
                DSO ORDERS_DSO
                  USING(2751,2761)
                  ALLOCATE PRIME    ON SP126
SIZE 41984K,
                        OVERFLOW ON SP126 SIZE
320K;

        CREATE DSI ORDERS_252_DSI
                DSO ORDERS_DSO
                  USING(2762,5544)
                  ALLOCATE PRIME    ON SP126
SIZE 41984K,
                        OVERFLOW ON SP126 SIZE
320K;

-------------------------------------------------------------------
---------
-- * Phase.2-4b: Orders-IX
-------------------------------------------------------------------
---------
        CREATE DSO ORDERS_IX_DSO
                INDEX ON
TPCC_SCHEMA.ORDERS(O_C_ID,O_W_ID,O_
D_ID)
                TYPE
BTREE(PAGESIZE1(4),PAGESIZE2(32));

        CREATE DSI ORDERS_IX_1_DSI
                INDEX
                DSO ORDERS_IX_DSO
                BASE ORDERS_1_DSI
                ALLOCATE INDEX    ON SP1 SIZE
512K,
                        BASE     ON SP1 SIZE 16448K;

        CREATE DSI ORDERS_IX_2_DSI
                INDEX
                DSO ORDERS_IX_DSO
                BASE ORDERS_2_DSI
                ALLOCATE INDEX    ON SP1 SIZE
512K,
                        BASE     ON SP1 SIZE 16448K;

    .
    .
    .

        CREATE DSI ORDERS_IX_251_DSI
                INDEX
                DSO ORDERS_IX_DSO
                BASE ORDERS_251_DSI
                ALLOCATE INDEX    ON SP126 SIZE
512K,
                        BASE     ON SP126 SIZE
16448K;

        CREATE DSI ORDERS_IX_252_DSI
                INDEX
                DSO ORDERS_IX_DSO
                BASE ORDERS_252_DSI
                ALLOCATE INDEX    ON SP126 SIZE
512K,
                        BASE     ON SP126 SIZE
16448K;

-------------------------------------------------------------------
---------
-- * Phase.2-7: History
-------------------------------------------------------------------
---------
        CREATE DSO HISTORY_DSO
                FROM TPCC_SCHEMA.HISTORY
                        TYPE
SEQUENTIAL(PAGESIZE(4),ORDER(0))
                        WHERE (H_W_ID) BETWEEN
(?) AND (?);

        CREATE DSI HISTORY_1_DSI
                DSO HISTORY_DSO
                  USING(1,11)
                  ALLOCATE DATA     ON SP1 SIZE
33408K;

        CREATE DSI HISTORY_2_DSI
                DSO HISTORY_DSO
                  USING(12,22)
                  ALLOCATE DATA     ON SP1 SIZE
33408K;

    .
    .
    .

        CREATE DSI HISTORY_251_DSI
                DSO HISTORY_DSO
                  USING(2751,2761)
                  ALLOCATE DATA     ON SP126 SIZE
33408K;

        CREATE DSI HISTORY_252_DSI
                DSO HISTORY_DSO
                  USING(2762,5544)
                  ALLOCATE DATA     ON SP126 SIZE
33408K;

-------------------------------------------------------------------
---------
-- * Phase.2-3a: Customer
-------------------------------------------------------------------
---------
        CREATE DSO CUSTOMER_DSO
                FROM TPCC_SCHEMA.CUSTOMER
                        TYPE
RANDOM(PAGESIZE1(4),PAGESIZE2(1),

        RULE(C_ID*22+C_W_ID*2+C_D_ID/6+(C_
D_ID-(C_D_ID/5*5))*66000))
                        WHERE (C_W_ID) BETWEEN
(?) AND (?);

        CREATE DSI CUSTOMER_1_DSI
                DSO CUSTOMER_DSO
                  USING(1,11)
                  ALLOCATE PRIME    ON SP1 SIZE
264004K,
                        OVERFLOW ON SP1 SIZE
13244K;

        CREATE DSI CUSTOMER_2_DSI
                DSO CUSTOMER_DSO
                  USING(12,22)
                  ALLOCATE PRIME    ON SP1 SIZE
264004K,
                        OVERFLOW ON SP1 SIZE
13244K;

    .
    .
    .

        CREATE DSI CUSTOMER_251_DSI
                DSO CUSTOMER_DSO
                  USING(2751,2761)
                  ALLOCATE PRIME    ON SP126
SIZE 264004K,
                        OVERFLOW ON SP126 SIZE
13244K;

        CREATE DSI CUSTOMER_252_DSI
                DSO CUSTOMER_DSO
                  USING(2762,5544)
```

```
            ALLOCATE PRIME    ON SP126
SIZE 264004K,
                  OVERFLOW ON SP126 SIZE
13244K;

-------------------------------------------------------------------
---------
-- * Phase.2-3b: Customer-Index
-------------------------------------------------------------------
---------
      CREATE DSO CUSTOMER_IX_DSO
            INDEX ON
TPCC_SCHEMA.CUSTOMER(C_W_ID,C_D_ID,
C_LAST)
            TYPE
BTREE(PAGESIZE1(8),PAGESIZE2(32));

      CREATE DSI CUSTOMER_X_1_DSI
            INDEX
            DSO CUSTOMER_IX_DSO
            BASE CUSTOMER_1_DSI
            ALLOCATE INDEX    ON SP1 SIZE
64K,
                  BASE     ON SP1 SIZE 8448K;

      CREATE DSI CUSTOMER_X_2_DSI
            INDEX
            DSO CUSTOMER_IX_DSO
            BASE CUSTOMER_2_DSI
            ALLOCATE INDEX    ON SP1 SIZE
64K,
                  BASE     ON SP1 SIZE 8448K;

   .
   .
   .

      CREATE DSI CUSTOMER_X_251_DSI
            INDEX
            DSO CUSTOMER_IX_DSO
            BASE CUSTOMER_251_DSI
            ALLOCATE INDEX    ON SP126 SIZE
64K,
                  BASE     ON SP126 SIZE
8448K;

      CREATE DSI CUSTOMER_X_252_DSI
            INDEX
            DSO CUSTOMER_IX_DSO
            BASE CUSTOMER_252_DSI
            ALLOCATE INDEX    ON SP126 SIZE
64K,
                  BASE     ON SP126 SIZE
8448K;

-------------------------------------------------------------------
---------
-- * Phase.2-5a: OrderLine
-------------------------------------------------------------------
---------
      CREATE DSO ORDERLINE_DSO
            FROM TPCC_SCHEMA.ORDERLINE
                  TYPE
RANDOM(PAGESIZE1(32),PAGESIZE2(8),RULE(
(OL_O_ID/30)*110+OL_W_ID*10+OL_D_ID+(OL_
NUMBER+(OL_O_ID-
((OL_O_ID/30)*30))*15)*13757))
                  WHERE (OL_W_ID) BETWEEN
(?) AND (?);

      CREATE DSI ORDERLIN_1_DSI
            DSO ORDERLINE_DSO
              USING(1,11)
              ALLOCATE PRIME    ON SP201
SIZE 440256K,
                  OVERFLOW ON SP201 SIZE
1728K;

      CREATE DSI ORDERLIN_2_DSI
            DSO ORDERLINE_DSO
              USING(12,22)
              ALLOCATE PRIME    ON SP202
SIZE 440256K,
                  OVERFLOW ON SP202 SIZE
1728K;

   .
   .
   .

      CREATE DSI ORDERLIN_251_DSI
            DSO ORDERLINE_DSO
              USING(2751,2761)
              ALLOCATE PRIME    ON SP263
SIZE 440256K,
                  OVERFLOW ON SP263 SIZE
1728K;

      CREATE DSI ORDERLIN_252_DSI
            DSO ORDERLINE_DSO
              USING(2762,5544)
              ALLOCATE PRIME    ON SP264
SIZE 440256K,
                  OVERFLOW ON SP264 SIZE
1728K;

-------------------------------------------------------------------
---------
-- * Phase.2-9: Item
-------------------------------------------------------------------
---------
      CREATE DSO ITEM_DSO
            FROM TPCC_SCHEMA.ITEM
                  TYPE
RANDOM(PAGESIZE1(1),PAGESIZE2(1),RULE(I
_ID/7+(I_ID-((I_ID/7)*7))*14286));

      CREATE DSI ITEM_1_DSI
            DSO ITEM_DSO
              ALLOCATE PRIME    ON      SP1
SIZE 1615K

      SP2 SIZE 1408K

      SP3 SIZE 1408K

      SP4 SIZE 1408K

      SP5 SIZE 1408K

      SP6 SIZE 1408K

      SP7 SIZE 1408K

      SP8 SIZE 1408K

      SP9 SIZE 1408K

      SP10 SIZE 1408K,
                  OVERFLOW ON  SP1 SIZE
753K;
```

File: ddl.dbsp.dat

```
CREATE DBSPACE SP1    ALLOCATE
RAWDEVICE /DEV/rdsk/SP1;
CREATE DBSPACE SP2    ALLOCATE
RAWDEVICE /DEV/rdsk/SP2;
CREATE DBSPACE SP3    ALLOCATE
RAWDEVICE /DEV/rdsk/SP3;

CREATE DBSPACE SP4    ALLOCATE
RAWDEVICE /DEV/rdsk/SP4;
CREATE DBSPACE SP5    ALLOCATE
RAWDEVICE /DEV/rdsk/SP5;
CREATE DBSPACE SP6    ALLOCATE
RAWDEVICE /DEV/rdsk/SP6;

CREATE DBSPACE SP7    ALLOCATE
RAWDEVICE /DEV/rdsk/SP7;
CREATE DBSPACE SP8    ALLOCATE
RAWDEVICE /DEV/rdsk/SP8;
CREATE DBSPACE SP9    ALLOCATE
RAWDEVICE /DEV/rdsk/SP9;

CREATE DBSPACE SP10   ALLOCATE
RAWDEVICE /DEV/rdsk/SP10;
CREATE DBSPACE SP11   ALLOCATE
RAWDEVICE /DEV/rdsk/SP11;
CREATE DBSPACE SP12   ALLOCATE
RAWDEVICE /DEV/rdsk/SP12;

CREATE DBSPACE SP13   ALLOCATE
RAWDEVICE /DEV/rdsk/SP13;
CREATE DBSPACE SP14   ALLOCATE
RAWDEVICE /DEV/rdsk/SP14;
CREATE DBSPACE SP15   ALLOCATE
RAWDEVICE /DEV/rdsk/SP15;

CREATE DBSPACE SP16   ALLOCATE
RAWDEVICE /DEV/rdsk/SP16;
CREATE DBSPACE SP17   ALLOCATE
RAWDEVICE /DEV/rdsk/SP17;
CREATE DBSPACE SP18   ALLOCATE
RAWDEVICE /DEV/rdsk/SP18;

CREATE DBSPACE SP19   ALLOCATE
RAWDEVICE /DEV/rdsk/SP19;
CREATE DBSPACE SP20   ALLOCATE
RAWDEVICE /DEV/rdsk/SP20;
CREATE DBSPACE SP21   ALLOCATE
RAWDEVICE /DEV/rdsk/SP21;

CREATE DBSPACE SP22   ALLOCATE
RAWDEVICE /DEV/rdsk/SP22;
CREATE DBSPACE SP23   ALLOCATE
RAWDEVICE /DEV/rdsk/SP23;
CREATE DBSPACE SP24   ALLOCATE
RAWDEVICE /DEV/rdsk/SP24;
```

```
CREATE DBSPACE SP25  ALLOCATE
RAWDEVICE /DEV/rdsk/SP25;
CREATE DBSPACE SP26  ALLOCATE
RAWDEVICE /DEV/rdsk/SP26;
CREATE DBSPACE SP27  ALLOCATE
RAWDEVICE /DEV/rdsk/SP27;

CREATE DBSPACE SP28  ALLOCATE
RAWDEVICE /DEV/rdsk/SP28;
CREATE DBSPACE SP29  ALLOCATE
RAWDEVICE /DEV/rdsk/SP29;
CREATE DBSPACE SP30  ALLOCATE
RAWDEVICE /DEV/rdsk/SP30;

CREATE DBSPACE SP31  ALLOCATE
RAWDEVICE /DEV/rdsk/SP31;
CREATE DBSPACE SP32  ALLOCATE
RAWDEVICE /DEV/rdsk/SP32;
CREATE DBSPACE SP33  ALLOCATE
RAWDEVICE /DEV/rdsk/SP33;

CREATE DBSPACE SP34  ALLOCATE
RAWDEVICE /DEV/rdsk/SP34;
CREATE DBSPACE SP35  ALLOCATE
RAWDEVICE /DEV/rdsk/SP35;
CREATE DBSPACE SP36  ALLOCATE
RAWDEVICE /DEV/rdsk/SP36;

CREATE DBSPACE SP37  ALLOCATE
RAWDEVICE /DEV/rdsk/SP37;
CREATE DBSPACE SP38  ALLOCATE
RAWDEVICE /DEV/rdsk/SP38;
CREATE DBSPACE SP39  ALLOCATE
RAWDEVICE /DEV/rdsk/SP39;

CREATE DBSPACE SP40  ALLOCATE
RAWDEVICE /DEV/rdsk/SP40;
CREATE DBSPACE SP41  ALLOCATE
RAWDEVICE /DEV/rdsk/SP41;
CREATE DBSPACE SP42  ALLOCATE
RAWDEVICE /DEV/rdsk/SP42;

CREATE DBSPACE SP43  ALLOCATE
RAWDEVICE /DEV/rdsk/SP43;
CREATE DBSPACE SP44  ALLOCATE
RAWDEVICE /DEV/rdsk/SP44;
CREATE DBSPACE SP45  ALLOCATE
RAWDEVICE /DEV/rdsk/SP45;

CREATE DBSPACE SP46  ALLOCATE
RAWDEVICE /DEV/rdsk/SP46;
CREATE DBSPACE SP47  ALLOCATE
RAWDEVICE /DEV/rdsk/SP47;
CREATE DBSPACE SP48  ALLOCATE
RAWDEVICE /DEV/rdsk/SP48;

CREATE DBSPACE SP49  ALLOCATE
RAWDEVICE /DEV/rdsk/SP49;
CREATE DBSPACE SP50  ALLOCATE
RAWDEVICE /DEV/rdsk/SP50;
CREATE DBSPACE SP51  ALLOCATE
RAWDEVICE /DEV/rdsk/SP51;

CREATE DBSPACE SP52  ALLOCATE
RAWDEVICE /DEV/rdsk/SP52;
CREATE DBSPACE SP53  ALLOCATE
RAWDEVICE /DEV/rdsk/SP53;

CREATE DBSPACE SP54  ALLOCATE
RAWDEVICE /DEV/rdsk/SP54;

CREATE DBSPACE SP55  ALLOCATE
RAWDEVICE /DEV/rdsk/SP55;
CREATE DBSPACE SP56  ALLOCATE
RAWDEVICE /DEV/rdsk/SP56;
CREATE DBSPACE SP57  ALLOCATE
RAWDEVICE /DEV/rdsk/SP57;

CREATE DBSPACE SP58  ALLOCATE
RAWDEVICE /DEV/rdsk/SP58;
CREATE DBSPACE SP59  ALLOCATE
RAWDEVICE /DEV/rdsk/SP59;
CREATE DBSPACE SP60  ALLOCATE
RAWDEVICE /DEV/rdsk/SP60;

CREATE DBSPACE SP61  ALLOCATE
RAWDEVICE /DEV/rdsk/SP61;
CREATE DBSPACE SP62  ALLOCATE
RAWDEVICE /DEV/rdsk/SP62;
CREATE DBSPACE SP63  ALLOCATE
RAWDEVICE /DEV/rdsk/SP63;

CREATE DBSPACE SP64  ALLOCATE
RAWDEVICE /DEV/rdsk/SP64;
CREATE DBSPACE SP65  ALLOCATE
RAWDEVICE /DEV/rdsk/SP65;
CREATE DBSPACE SP66  ALLOCATE
RAWDEVICE /DEV/rdsk/SP66;

CREATE DBSPACE SP67  ALLOCATE
RAWDEVICE /DEV/rdsk/SP67;
CREATE DBSPACE SP68  ALLOCATE
RAWDEVICE /DEV/rdsk/SP68;
CREATE DBSPACE SP69  ALLOCATE
RAWDEVICE /DEV/rdsk/SP69;

CREATE DBSPACE SP70  ALLOCATE
RAWDEVICE /DEV/rdsk/SP70;
CREATE DBSPACE SP71  ALLOCATE
RAWDEVICE /DEV/rdsk/SP71;
CREATE DBSPACE SP72  ALLOCATE
RAWDEVICE /DEV/rdsk/SP72;

CREATE DBSPACE SP73  ALLOCATE
RAWDEVICE /DEV/rdsk/SP73;
CREATE DBSPACE SP74  ALLOCATE
RAWDEVICE /DEV/rdsk/SP74;
CREATE DBSPACE SP75  ALLOCATE
RAWDEVICE /DEV/rdsk/SP75;

CREATE DBSPACE SP76  ALLOCATE
RAWDEVICE /DEV/rdsk/SP76;
CREATE DBSPACE SP77  ALLOCATE
RAWDEVICE /DEV/rdsk/SP77;
CREATE DBSPACE SP78  ALLOCATE
RAWDEVICE /DEV/rdsk/SP78;

CREATE DBSPACE SP79  ALLOCATE
RAWDEVICE /DEV/rdsk/SP79;
CREATE DBSPACE SP80  ALLOCATE
RAWDEVICE /DEV/rdsk/SP80;
CREATE DBSPACE SP81  ALLOCATE
RAWDEVICE /DEV/rdsk/SP81;

CREATE DBSPACE SP82  ALLOCATE
RAWDEVICE /DEV/rdsk/SP82;

CREATE DBSPACE SP83  ALLOCATE
RAWDEVICE /DEV/rdsk/SP83;
CREATE DBSPACE SP84  ALLOCATE
RAWDEVICE /DEV/rdsk/SP84;

CREATE DBSPACE SP85  ALLOCATE
RAWDEVICE /DEV/rdsk/SP85;
CREATE DBSPACE SP86  ALLOCATE
RAWDEVICE /DEV/rdsk/SP86;
CREATE DBSPACE SP87  ALLOCATE
RAWDEVICE /DEV/rdsk/SP87;

CREATE DBSPACE SP88  ALLOCATE
RAWDEVICE /DEV/rdsk/SP88;
CREATE DBSPACE SP89  ALLOCATE
RAWDEVICE /DEV/rdsk/SP89;
CREATE DBSPACE SP90  ALLOCATE
RAWDEVICE /DEV/rdsk/SP90;

CREATE DBSPACE SP91  ALLOCATE
RAWDEVICE /DEV/rdsk/SP91;
CREATE DBSPACE SP92  ALLOCATE
RAWDEVICE /DEV/rdsk/SP92;
CREATE DBSPACE SP93  ALLOCATE
RAWDEVICE /DEV/rdsk/SP93;

CREATE DBSPACE SP94  ALLOCATE
RAWDEVICE /DEV/rdsk/SP94;
CREATE DBSPACE SP95  ALLOCATE
RAWDEVICE /DEV/rdsk/SP95;
CREATE DBSPACE SP96  ALLOCATE
RAWDEVICE /DEV/rdsk/SP96;

CREATE DBSPACE SP97  ALLOCATE
RAWDEVICE /DEV/rdsk/SP97;
CREATE DBSPACE SP98  ALLOCATE
RAWDEVICE /DEV/rdsk/SP98;
CREATE DBSPACE SP99  ALLOCATE
RAWDEVICE /DEV/rdsk/SP99;

CREATE DBSPACE SP100  ALLOCATE
RAWDEVICE /DEV/rdsk/SP100;
CREATE DBSPACE SP101  ALLOCATE
RAWDEVICE /DEV/rdsk/SP101;
CREATE DBSPACE SP102  ALLOCATE
RAWDEVICE /DEV/rdsk/SP102;

CREATE DBSPACE SP103  ALLOCATE
RAWDEVICE /DEV/rdsk/SP103;
CREATE DBSPACE SP104  ALLOCATE
RAWDEVICE /DEV/rdsk/SP104;
CREATE DBSPACE SP105  ALLOCATE
RAWDEVICE /DEV/rdsk/SP105;

CREATE DBSPACE SP106  ALLOCATE
RAWDEVICE /DEV/rdsk/SP106;
CREATE DBSPACE SP107  ALLOCATE
RAWDEVICE /DEV/rdsk/SP107;
CREATE DBSPACE SP108  ALLOCATE
RAWDEVICE /DEV/rdsk/SP108;

CREATE DBSPACE SP109  ALLOCATE
RAWDEVICE /DEV/rdsk/SP109;
CREATE DBSPACE SP110  ALLOCATE
RAWDEVICE /DEV/rdsk/SP110;
CREATE DBSPACE SP111  ALLOCATE
RAWDEVICE /DEV/rdsk/SP111;
```

```
CREATE DBSPACE SP112  ALLOCATE
RAWDEVICE /DEV/rdsk/SP112;
CREATE DBSPACE SP113  ALLOCATE
RAWDEVICE /DEV/rdsk/SP113;
CREATE DBSPACE SP114  ALLOCATE
RAWDEVICE /DEV/rdsk/SP114;


CREATE DBSPACE SP115  ALLOCATE
RAWDEVICE /DEV/rdsk/SP115;
CREATE DBSPACE SP116  ALLOCATE
RAWDEVICE /DEV/rdsk/SP116;
CREATE DBSPACE SP117  ALLOCATE
RAWDEVICE /DEV/rdsk/SP117;

CREATE DBSPACE SP118  ALLOCATE
RAWDEVICE /DEV/rdsk/SP118;
CREATE DBSPACE SP119  ALLOCATE
RAWDEVICE /DEV/rdsk/SP119;
CREATE DBSPACE SP120  ALLOCATE
RAWDEVICE /DEV/rdsk/SP120;

CREATE DBSPACE SP121  ALLOCATE
RAWDEVICE /DEV/rdsk/SP121;
CREATE DBSPACE SP122  ALLOCATE
RAWDEVICE /DEV/rdsk/SP122;
CREATE DBSPACE SP123  ALLOCATE
RAWDEVICE /DEV/rdsk/SP123;

CREATE DBSPACE SP124  ALLOCATE
RAWDEVICE /DEV/rdsk/SP124;
CREATE DBSPACE SP125  ALLOCATE
RAWDEVICE /DEV/rdsk/SP125;
CREATE DBSPACE SP126  ALLOCATE
RAWDEVICE /DEV/rdsk/SP126;


CREATE DBSPACE SP201  ALLOCATE
RAWDEVICE /DEV/rdsk/SP201;
CREATE DBSPACE SP202  ALLOCATE
RAWDEVICE /DEV/rdsk/SP202;
CREATE DBSPACE SP203  ALLOCATE
RAWDEVICE /DEV/rdsk/SP203;
CREATE DBSPACE SP204  ALLOCATE
RAWDEVICE /DEV/rdsk/SP204;
CREATE DBSPACE SP205  ALLOCATE
RAWDEVICE /DEV/rdsk/SP205;

CREATE DBSPACE SP206  ALLOCATE
RAWDEVICE /DEV/rdsk/SP206;
CREATE DBSPACE SP207  ALLOCATE
RAWDEVICE /DEV/rdsk/SP207;
CREATE DBSPACE SP208  ALLOCATE
RAWDEVICE /DEV/rdsk/SP208;
CREATE DBSPACE SP209  ALLOCATE
RAWDEVICE /DEV/rdsk/SP209;
CREATE DBSPACE SP210  ALLOCATE
RAWDEVICE /DEV/rdsk/SP210;

CREATE DBSPACE SP211  ALLOCATE
RAWDEVICE /DEV/rdsk/SP211;
CREATE DBSPACE SP212  ALLOCATE
RAWDEVICE /DEV/rdsk/SP212;
CREATE DBSPACE SP213  ALLOCATE
RAWDEVICE /DEV/rdsk/SP213;
CREATE DBSPACE SP214  ALLOCATE
RAWDEVICE /DEV/rdsk/SP214;

CREATE DBSPACE SP215  ALLOCATE
RAWDEVICE /DEV/rdsk/SP215;

CREATE DBSPACE SP216  ALLOCATE
RAWDEVICE /DEV/rdsk/SP216;
CREATE DBSPACE SP217  ALLOCATE
RAWDEVICE /DEV/rdsk/SP217;
CREATE DBSPACE SP218  ALLOCATE
RAWDEVICE /DEV/rdsk/SP218;
CREATE DBSPACE SP219  ALLOCATE
RAWDEVICE /DEV/rdsk/SP219;
CREATE DBSPACE SP220  ALLOCATE
RAWDEVICE /DEV/rdsk/SP220;

CREATE DBSPACE SP221  ALLOCATE
RAWDEVICE /DEV/rdsk/SP221;
CREATE DBSPACE SP222  ALLOCATE
RAWDEVICE /DEV/rdsk/SP222;
CREATE DBSPACE SP223  ALLOCATE
RAWDEVICE /DEV/rdsk/SP223;
CREATE DBSPACE SP224  ALLOCATE
RAWDEVICE /DEV/rdsk/SP224;
CREATE DBSPACE SP225  ALLOCATE
RAWDEVICE /DEV/rdsk/SP225;

CREATE DBSPACE SP226  ALLOCATE
RAWDEVICE /DEV/rdsk/SP226;
CREATE DBSPACE SP227  ALLOCATE
RAWDEVICE /DEV/rdsk/SP227;
CREATE DBSPACE SP228  ALLOCATE
RAWDEVICE /DEV/rdsk/SP228;
CREATE DBSPACE SP229  ALLOCATE
RAWDEVICE /DEV/rdsk/SP229;
CREATE DBSPACE SP230  ALLOCATE
RAWDEVICE /DEV/rdsk/SP230;

CREATE DBSPACE SP231  ALLOCATE
RAWDEVICE /DEV/rdsk/SP231;
CREATE DBSPACE SP232  ALLOCATE
RAWDEVICE /DEV/rdsk/SP232;
CREATE DBSPACE SP233  ALLOCATE
RAWDEVICE /DEV/rdsk/SP233;
CREATE DBSPACE SP234  ALLOCATE
RAWDEVICE /DEV/rdsk/SP234;

CREATE DBSPACE SP235  ALLOCATE
RAWDEVICE /DEV/rdsk/SP235;
CREATE DBSPACE SP236  ALLOCATE
RAWDEVICE /DEV/rdsk/SP236;
CREATE DBSPACE SP237  ALLOCATE
RAWDEVICE /DEV/rdsk/SP237;
CREATE DBSPACE SP238  ALLOCATE
RAWDEVICE /DEV/rdsk/SP238;


CREATE DBSPACE SP239  ALLOCATE
RAWDEVICE /DEV/rdsk/SP239;
CREATE DBSPACE SP240  ALLOCATE
RAWDEVICE /DEV/rdsk/SP240;
CREATE DBSPACE SP241  ALLOCATE
RAWDEVICE /DEV/rdsk/SP241;
CREATE DBSPACE SP242  ALLOCATE
RAWDEVICE /DEV/rdsk/SP242;
CREATE DBSPACE SP243  ALLOCATE
RAWDEVICE /DEV/rdsk/SP243;

CREATE DBSPACE SP244  ALLOCATE
RAWDEVICE /DEV/rdsk/SP244;

CREATE DBSPACE SP245  ALLOCATE
RAWDEVICE /DEV/rdsk/SP245;
CREATE DBSPACE SP246  ALLOCATE
RAWDEVICE /DEV/rdsk/SP246;
CREATE DBSPACE SP247  ALLOCATE
RAWDEVICE /DEV/rdsk/SP247;
CREATE DBSPACE SP248  ALLOCATE
RAWDEVICE /DEV/rdsk/SP248;

CREATE DBSPACE SP249  ALLOCATE
RAWDEVICE /DEV/rdsk/SP249;
CREATE DBSPACE SP250  ALLOCATE
RAWDEVICE /DEV/rdsk/SP250;
CREATE DBSPACE SP251  ALLOCATE
RAWDEVICE /DEV/rdsk/SP251;
CREATE DBSPACE SP252  ALLOCATE
RAWDEVICE /DEV/rdsk/SP252;
CREATE DBSPACE SP253  ALLOCATE
RAWDEVICE /DEV/rdsk/SP253;

CREATE DBSPACE SP254  ALLOCATE
RAWDEVICE /DEV/rdsk/SP254;
CREATE DBSPACE SP255  ALLOCATE
RAWDEVICE /DEV/rdsk/SP255;
CREATE DBSPACE SP256  ALLOCATE
RAWDEVICE /DEV/rdsk/SP256;
CREATE DBSPACE SP257  ALLOCATE
RAWDEVICE /DEV/rdsk/SP257;
CREATE DBSPACE SP258  ALLOCATE
RAWDEVICE /DEV/rdsk/SP258;

CREATE DBSPACE SP259  ALLOCATE
RAWDEVICE /DEV/rdsk/SP259;
CREATE DBSPACE SP260  ALLOCATE
RAWDEVICE /DEV/rdsk/SP260;
CREATE DBSPACE SP261  ALLOCATE
RAWDEVICE /DEV/rdsk/SP261;
CREATE DBSPACE SP262  ALLOCATE
RAWDEVICE /DEV/rdsk/SP262;
CREATE DBSPACE SP263  ALLOCATE
RAWDEVICE /DEV/rdsk/SP263;

CREATE DBSPACE SP264  ALLOCATE
RAWDEVICE /DEV/rdsk/SP264;
CREATE DBSPACE SP265  ALLOCATE
RAWDEVICE /DEV/rdsk/SP265;
CREATE DBSPACE SP266  ALLOCATE
RAWDEVICE /DEV/rdsk/SP266;
CREATE DBSPACE SP267  ALLOCATE
RAWDEVICE /DEV/rdsk/SP267;
CREATE DBSPACE SP268  ALLOCATE
RAWDEVICE /DEV/rdsk/SP268;

CREATE DBSPACE SP269  ALLOCATE
RAWDEVICE /DEV/rdsk/SP269;
CREATE DBSPACE SP270  ALLOCATE
RAWDEVICE /DEV/rdsk/SP270;
CREATE DBSPACE SP271  ALLOCATE
RAWDEVICE /DEV/rdsk/SP271;
```

File: ddl_db.mak

```
--
/*====================================
==========================*/
```

```
--
/*====================================
==========================*/
  CREATE DATABASE TPCC;
```

File: LOAD.sh

```
#!/bin/csh -xf

setenv RDBDB TPCC

set LOAD1_D = /rdb/loaddata/1
set LOAD2_D = /rdb/loaddata/2
set LOAD3_D = /rdb/loaddata/3
set LOAD4_D = /rdb/loaddata/4
set LOAD5_D = /rdb/loaddata/5
set LOAD6_D = /rdb/loaddata/6
set LOAD7_D = /rdb/loaddata/7
set LOAD8_D = /rdb/loaddata/8
set LOAD9_D = /rdb/loaddata/9
set LOAD10_D = /rdb/loaddata/10
set LOAD11_D = /rdb/loaddata/11
set LOAD12_D = /rdb/loaddata/12
set LOAD13_D = /rdb/loaddata/13
set LOAD14_D = /rdb/loaddata/14
set LOAD15_D = /rdb/loaddata/15
set LOAD16_D = /rdb/loaddata/16
set WK1_D = /rdb/sortwk1
set WK2_D = /rdb/sortwk2
set WK3_D = /rdb/sortwk3
set WK4_D = /rdb/sortwk4
set WK5_D = /rdb/sortwk5
set WK6_D = /rdb/sortwk6
set WK7_D = /rdb/sortwk7
set WK8_D = /rdb/sortwk8
set WK9_D = /rdb/sortwk9
set WK10_D = /rdb/sortwk10
set WK11_D = /rdb/sortwk11
set WK12_D = /rdb/sortwk12
set WK13_D = /rdb/sortwk13
set WK14_D = /rdb/sortwk14
set WK15_D = /rdb/sortwk15
set WK16_D = /rdb/sortwk16

rm /rdb/loaddata/*/??*_* /rdb/loaddata/*/data
rm /rdb/sortwk*/SRT*

## Item

        wttpccd1 $LOAD1_D 1 1 I
        timex rdbsloader -mi -i
$RDBDB.ITEM_1_DSI \
                                                -s
$WK1_D \
                                                -s
$WK2_D \
                                                -s
$WK3_D \
                                                -s
$WK4_D \
                                                -
n $LOAD1_D/data
rm $LOAD1_D/data

## WAREHOUSE

        @ make_s1 = 1
        @ make_e1 = 66
        @ make_s2 = 67
        @ make_e2 = 132
        @ make_s3 = 133
        @ make_e3 = 198
        @ make_s4 = 199
        @ make_e4 = 264
        @ make_s5 = 265
        @ make_e5 = 330
        @ make_s6 = 331
        @ make_e6 = 396
        @ make_s7 = 397
        @ make_e7 = 462
        @ make_s8 = 463
        @ make_e8 = 528

        wttpccd1 $LOAD1_D $make_s1 $make_e1
W &
        wttpccd1 $LOAD2_D $make_s2 $make_e2
W &
        wttpccd1 $LOAD3_D $make_s3 $make_e3
W &
        wttpccd1 $LOAD4_D $make_s4 $make_e4
W &
        wttpccd1 $LOAD5_D $make_s5 $make_e5
W &
        wttpccd1 $LOAD6_D $make_s6 $make_e6
W &
        wttpccd1 $LOAD7_D $make_s7 $make_e7
W &
        wttpccd1 $LOAD8_D $make_s8 $make_e8
W &

        wait


foreach num ( 0 1 2 3 4 5 )


        @ dsi_num1 = $num * 8 + 1
        @ dsi_num2 = $num * 8 + 2
        @ dsi_num3 = $num * 8 + 3
        @ dsi_num4 = $num * 8 + 4
        @ dsi_num5 = $num * 8 + 5
        @ dsi_num6 = $num * 8 + 6
        @ dsi_num7 = $num * 8 + 7
        @ dsi_num8 = $num * 8 + 8
        timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num1\_DSI \
                                                -s
$WK1_D \
                                                -s
$WK2_D \
                                                -
n $LOAD1_D/WH$make_s1\_$make_e1 &
        timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num2\_DSI \
                                                -s
$WK3_D \
                                                -s
$WK4_D \
                                                -
n $LOAD2_D/WH$make_s2\_$make_e2 &
        timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num3\_DSI \
                                                -s
$WK5_D \
                                                -s
$WK6_D \
                                                -
n $LOAD3_D/WH$make_s3\_$make_e3 &
        timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num4\_DSI \
                                                -s
$WK7_D \
                                                -s
$WK8_D \
                                                -
n $LOAD4_D/WH$make_s4\_$make_e4 &
        timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num5\_DSI \
                                                -s
$WK9_D \
                                                -s
$WK10_D \
                                                -
n $LOAD5_D/WH$make_s5\_$make_e5 &
        timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num6\_DSI \
                                                -s
$WK11_D \
                                                -s
$WK12_D \
                                                -
n $LOAD6_D/WH$make_s6\_$make_e6 &
        timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num7\_DSI \
                                                -s
$WK13_D \
                                                -s
$WK14_D \
                                                -
n $LOAD7_D/WH$make_s7\_$make_e7 &
        timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num8\_DSI \
                                                -s
$WK15_D \
                                                -s
$WK16_D \
                                                -
n $LOAD8_D/WH$make_s8\_$make_e8 &


        @ rm_make_s1 = $make_s1
        @ rm_make_e1 = $make_e1
        @ rm_make_s2 = $make_s2
        @ rm_make_e2 = $make_e2
        @ rm_make_s3 = $make_s3
        @ rm_make_e3 = $make_e3
        @ rm_make_s4 = $make_s4
        @ rm_make_e4 = $make_e4
        @ rm_make_s5 = $make_s5
        @ rm_make_e5 = $make_e5
        @ rm_make_s6 = $make_s6
        @ rm_make_e6 = $make_e6
        @ rm_make_s7 = $make_s7
        @ rm_make_e7 = $make_e7
        @ rm_make_s8 = $make_s8
        @ rm_make_e8 = $make_e8

        @ make_s1 = $num * 528 + 529
        @ make_e1 = $num * 528 + 594
        @ make_s2 = $num * 528 + 595
```

```
        @ make_e2 = $num * 528 + 660
        @ make_s3 = $num * 528 + 661
        @ make_e3 = $num * 528 + 726
        @ make_s4 = $num * 528 + 727
        @ make_e4 = $num * 528 + 792
        @ make_s5 = $num * 528 + 793
        @ make_e5 = $num * 528 + 858
        @ make_s6 = $num * 528 + 859
        @ make_e6 = $num * 528 + 924
        @ make_s7 = $num * 528 + 925
        @ make_e7 = $num * 528 + 990
        @ make_s8 = $num * 528 + 991
        @ make_e8 = $num * 528 + 1056

        wttpccd1 $LOAD1_D $make_s1 $make_e1
W &
        wttpccd1 $LOAD2_D $make_s2 $make_e2
W &
        wttpccd1 $LOAD3_D $make_s3 $make_e3
W &
        wttpccd1 $LOAD4_D $make_s4 $make_e4
W &
        wttpccd1 $LOAD5_D $make_s5 $make_e5
W &
        wttpccd1 $LOAD6_D $make_s6 $make_e6
W &
        wttpccd1 $LOAD7_D $make_s7 $make_e7
W &
        wttpccd1 $LOAD8_D $make_s8 $make_e8
W &

        wait

        rm
$LOAD1_D/WH$rm_make_s1\_$rm_make_e1
        rm
$LOAD2_D/WH$rm_make_s2\_$rm_make_e2
        rm
$LOAD3_D/WH$rm_make_s3\_$rm_make_e3
        rm
$LOAD4_D/WH$rm_make_s4\_$rm_make_e4
        rm
$LOAD5_D/WH$rm_make_s5\_$rm_make_e5
        rm
$LOAD6_D/WH$rm_make_s6\_$rm_make_e6
        rm
$LOAD7_D/WH$rm_make_s7\_$rm_make_e7
        rm
$LOAD8_D/WH$rm_make_s8\_$rm_make_e8

end

        rm $LOAD1_D/WH$make_s1\_$make_e1
        rm $LOAD2_D/WH$make_s2\_$make_e2
        rm $LOAD3_D/WH$make_s3\_$make_e3
        rm $LOAD4_D/WH$make_s4\_$make_e4
        rm $LOAD5_D/WH$make_s5\_$make_e5
        rm $LOAD6_D/WH$make_s6\_$make_e6
        rm $LOAD7_D/WH$make_s7\_$make_e7
        rm $LOAD8_D/WH$make_s8\_$make_e8



## DISTRICT

        @ make_s1 = 1
        @ make_e1 = 66
        @ make_s2 = 67
        @ make_e2 = 132
        @ make_s3 = 133
        @ make_e3 = 198
        @ make_s4 = 199
        @ make_e4 = 264
        @ make_s5 = 265
        @ make_e5 = 330
        @ make_s6 = 331
        @ make_e6 = 396
        @ make_s7 = 397
        @ make_e7 = 462
        @ make_s8 = 463
        @ make_e8 = 528

        wttpccd1 $LOAD1_D $make_s1 $make_e1
D &
        wttpccd1 $LOAD2_D $make_s2 $make_e2
D &
        wttpccd1 $LOAD3_D $make_s3 $make_e3
D &
        wttpccd1 $LOAD4_D $make_s4 $make_e4
D &
        wttpccd1 $LOAD5_D $make_s5 $make_e5
D &
        wttpccd1 $LOAD6_D $make_s6 $make_e6
D &
        wttpccd1 $LOAD7_D $make_s7 $make_e7
D &
        wttpccd1 $LOAD8_D $make_s8 $make_e8
D &

    wait

foreach num ( 0 1 2 3 4 5 )

        @ dsi_num1 = $num * 8 + 1
        @ dsi_num2 = $num * 8 + 2
        @ dsi_num3 = $num * 8 + 3
        @ dsi_num4 = $num * 8 + 4
        @ dsi_num5 = $num * 8 + 5
        @ dsi_num6 = $num * 8 + 6
        @ dsi_num7 = $num * 8 + 7
        @ dsi_num8 = $num * 8 + 8
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num1\_DSI \
                                                -s
$WK1_D \
                                                -s
$WK2_D \
                                                -
n $LOAD1_D/DI$make_s1\_$make_e1 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num2\_DSI \
                                                -s
$WK3_D \
                                                -s
$WK4_D \
                                                -
n $LOAD2_D/DI$make_s2\_$make_e2 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num3\_DSI \
                                                -s
$WK5_D \
                                                -s
$WK6_D \
                                                -
n $LOAD3_D/DI$make_s3\_$make_e3 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num4\_DSI \
                                                -s
$WK7_D \
                                                -s
$WK8_D \
                                                -
n $LOAD4_D/DI$make_s4\_$make_e4 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num5\_DSI \
                                                -s
$WK9_D \
                                                -s
$WK10_D \
                                                -
n $LOAD5_D/DI$make_s5\_$make_e5 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num6\_DSI \
                                                -s
$WK11_D \
                                                -s
$WK12_D \
                                                -
n $LOAD6_D/DI$make_s6\_$make_e6 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num7\_DSI \
                                                -s
$WK13_D \
                                                -s
$WK14_D \
                                                -
n $LOAD7_D/DI$make_s7\_$make_e7 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num8\_DSI \
                                                -s
$WK15_D \
                                                -s
$WK16_D \
                                                -
n $LOAD8_D/DI$make_s8\_$make_e8 &

        @ rm_make_s1 = $make_s1
        @ rm_make_e1 = $make_e1
        @ rm_make_s2 = $make_s2
        @ rm_make_e2 = $make_e2
        @ rm_make_s3 = $make_s3
        @ rm_make_e3 = $make_e3
        @ rm_make_s4 = $make_s4
        @ rm_make_e4 = $make_e4
        @ rm_make_s5 = $make_s5
        @ rm_make_e5 = $make_e5
        @ rm_make_s6 = $make_s6
        @ rm_make_e6 = $make_e6
        @ rm_make_s7 = $make_s7
        @ rm_make_e7 = $make_e7
        @ rm_make_s8 = $make_s8
        @ rm_make_e8 = $make_e8

        @ make_s1 = $num * 528 + 529
        @ make_e1 = $num * 528 + 594
        @ make_s2 = $num * 528 + 595
        @ make_e2 = $num * 528 + 660
        @ make_s3 = $num * 528 + 661
        @ make_e3 = $num * 528 + 726
        @ make_s4 = $num * 528 + 727
        @ make_e4 = $num * 528 + 792
        @ make_s5 = $num * 528 + 793
        @ make_e5 = $num * 528 + 858
```

```
        @ make_s6 = $num * 528 + 859
        @ make_e6 = $num * 528 + 924
        @ make_s7 = $num * 528 + 925
        @ make_e7 = $num * 528 + 990
        @ make_s8 = $num * 528 + 991
        @ make_e8 = $num * 528 + 1056

        wttpccd1 $LOAD1_D $make_s1 $make_e1
D &
        wttpccd1 $LOAD2_D $make_s2 $make_e2
D &
        wttpccd1 $LOAD3_D $make_s3 $make_e3
D &
        wttpccd1 $LOAD4_D $make_s4 $make_e4
D &
        wttpccd1 $LOAD5_D $make_s5 $make_e5
D &
        wttpccd1 $LOAD6_D $make_s6 $make_e6
D &
        wttpccd1 $LOAD7_D $make_s7 $make_e7
D &
        wttpccd1 $LOAD8_D $make_s8 $make_e8
D &

        wait

        rm
$LOAD1_D/DI$rm_make_s1\_$rm_make_e1
        rm
$LOAD2_D/DI$rm_make_s2\_$rm_make_e2
        rm
$LOAD3_D/DI$rm_make_s3\_$rm_make_e3
        rm
$LOAD4_D/DI$rm_make_s4\_$rm_make_e4
        rm
$LOAD5_D/DI$rm_make_s5\_$rm_make_e5
        rm
$LOAD6_D/DI$rm_make_s6\_$rm_make_e6
        rm
$LOAD7_D/DI$rm_make_s7\_$rm_make_e7
        rm
$LOAD8_D/DI$rm_make_s8\_$rm_make_e8

end

        rm $LOAD1_D/DI$make_s1\_$make_e1
        rm $LOAD2_D/DI$make_s2\_$make_e2
        rm $LOAD3_D/DI$make_s3\_$make_e3
        rm $LOAD4_D/DI$make_s4\_$make_e4
        rm $LOAD5_D/DI$make_s5\_$make_e5
        rm $LOAD6_D/DI$make_s6\_$make_e6
        rm $LOAD7_D/DI$make_s7\_$make_e7
        rm $LOAD8_D/DI$make_s8\_$make_e8

## CUSTOMER


        @ make_s1 = 1
        @ make_e1 = 11
        @ make_s2 = 12
        @ make_e2 = 22
        @ make_s3 = 23
        @ make_e3 = 33
        @ make_s4 = 34
        @ make_e4 = 44
        @ make_s5 = 45
        @ make_e5 = 55
        @ make_s6 = 56
        @ make_e6 = 66
        @ make_s7 = 67
        @ make_e7 = 77
        @ make_s8 = 78
        @ make_e8 = 88

        wttpccd1 $LOAD1_D $make_s1 $make_e1
C &
        wttpccd1 $LOAD2_D $make_s2 $make_e2
C &
        wttpccd1 $LOAD3_D $make_s3 $make_e3
C &
        wttpccd1 $LOAD4_D $make_s4 $make_e4
C &
        wttpccd1 $LOAD5_D $make_s5 $make_e5
C &
        wttpccd1 $LOAD6_D $make_s6 $make_e6
C &
        wttpccd1 $LOAD7_D $make_s7 $make_e7
C &
        wttpccd1 $LOAD8_D $make_s8 $make_e8
C &

        wait

foreach num ( 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
31 )

        @ dsi_num1 = $num * 8 + 1
        @ dsi_num2 = $num * 8 + 2
        @ dsi_num3 = $num * 8 + 3
        @ dsi_num4 = $num * 8 + 4
        @ dsi_num5 = $num * 8 + 5
        @ dsi_num6 = $num * 8 + 6
        @ dsi_num7 = $num * 8 + 7
        @ dsi_num8 = $num * 8 + 8

        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num1\_DSI -h \
                                        -s
$WK1_D \
                                        -s
$WK2_D \
                                        -
n $LOAD1_D/CU$make_s1\_$make_e1 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num2\_DSI -h \
                                        -s
$WK3_D \
                                        -s
$WK4_D \
                                        -
n $LOAD2_D/CU$make_s2\_$make_e2 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num3\_DSI -h \
                                        -s
$WK5_D \
                                        -s
$WK6_D \
                                        -
n $LOAD3_D/CU$make_s3\_$make_e3 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num4\_DSI -h \
                                        -s
$WK7_D \
                                        -s
$WK8_D \
                                        -
n $LOAD4_D/CU$make_s4\_$make_e4 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num5\_DSI -h \
                                        -s
$WK9_D \
                                        -s
$WK10_D \
                                        -
n $LOAD5_D/CU$make_s5\_$make_e5 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num6\_DSI -h \
                                        -s
$WK11_D \
                                        -s
$WK12_D \
                                        -
n $LOAD6_D/CU$make_s6\_$make_e6 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num7\_DSI -h \
                                        -s
$WK13_D \
                                        -s
$WK14_D \
                                        -
n $LOAD7_D/CU$make_s7\_$make_e7 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num8\_DSI -h \
                                        -s
$WK15_D \
                                        -s
$WK16_D \
                                        -
n $LOAD8_D/CU$make_s8\_$make_e8 &
        @ rm_make_s1 = $make_s1
        @ rm_make_e1 = $make_e1
        @ rm_make_s2 = $make_s2
        @ rm_make_e2 = $make_e2
        @ rm_make_s3 = $make_s3
        @ rm_make_e3 = $make_e3
        @ rm_make_s4 = $make_s4
        @ rm_make_e4 = $make_e4
        @ rm_make_s5 = $make_s5
        @ rm_make_e5 = $make_e5
        @ rm_make_s6 = $make_s6
        @ rm_make_e6 = $make_e6
        @ rm_make_s7 = $make_s7
        @ rm_make_e7 = $make_e7
        @ rm_make_s8 = $make_s8
        @ rm_make_e8 = $make_e8


        @ make_s1 = $num * 88 + 89
        @ make_e1 = $num * 88 + 99
        @ make_s2 = $num * 88 + 100
        @ make_e2 = $num * 88 + 110
        @ make_s3 = $num * 88 + 111
        @ make_e3 = $num * 88 + 121
        @ make_s4 = $num * 88 + 122
        @ make_e4 = $num * 88 + 132
        @ make_s5 = $num * 88 + 133
        @ make_e5 = $num * 88 + 143
        @ make_s6 = $num * 88 + 144
        @ make_e6 = $num * 88 + 154
        @ make_s7 = $num * 88 + 155
        @ make_e7 = $num * 88 + 165
        @ make_s8 = $num * 88 + 166
        @ make_e8 = $num * 88 + 176
```

```
        wttpccd1 $LOAD1_D $make_s1 $make_e1
C &
        wttpccd1 $LOAD2_D $make_s2 $make_e2
C &
        wttpccd1 $LOAD3_D $make_s3 $make_e3
C &
        wttpccd1 $LOAD4_D $make_s4 $make_e4
C &
        wttpccd1 $LOAD5_D $make_s5 $make_e5
C &
        wttpccd1 $LOAD6_D $make_s6 $make_e6
C &
        wttpccd1 $LOAD7_D $make_s7 $make_e7
C &
        wttpccd1 $LOAD8_D $make_s8 $make_e8
C &

        wait

    rm
$LOAD1_D/CU$rm_make_s1\_$rm_make_e1
    rm
$LOAD2_D/CU$rm_make_s2\_$rm_make_e2
    rm
$LOAD3_D/CU$rm_make_s3\_$rm_make_e3
    rm
$LOAD4_D/CU$rm_make_s4\_$rm_make_e4
    rm
$LOAD5_D/CU$rm_make_s5\_$rm_make_e5
    rm
$LOAD6_D/CU$rm_make_s6\_$rm_make_e6
    rm
$LOAD7_D/CU$rm_make_s7\_$rm_make_e7
    rm
$LOAD8_D/CU$rm_make_s8\_$rm_make_e8

end

    rm $LOAD1_D/CU$make_s1\_$make_e1
    rm $LOAD2_D/CU$make_s2\_$make_e2
    rm $LOAD3_D/CU$make_s3\_$make_e3
    rm $LOAD4_D/CU$make_s4\_$make_e4
    rm $LOAD5_D/CU$make_s5\_$make_e5
    rm $LOAD6_D/CU$make_s6\_$make_e6
    rm $LOAD7_D/CU$make_s7\_$make_e7
    rm $LOAD8_D/CU$make_s8\_$make_e8

## HISTORY

        @ make_s1 = 1
        @ make_e1 = 11
        @ make_s2 = 12
        @ make_e2 = 22
        @ make_s3 = 23
        @ make_e3 = 33
        @ make_s4 = 34
        @ make_e4 = 44
        @ make_s5 = 45
        @ make_e5 = 55
        @ make_s6 = 56
        @ make_e6 = 66
        @ make_s7 = 67
        @ make_e7 = 77
        @ make_s8 = 78
        @ make_e8 = 88

        wttpccd1 $LOAD1_D $make_s1 $make_e1
H &
        wttpccd1 $LOAD2_D $make_s2 $make_e2
H &
        wttpccd1 $LOAD3_D $make_s3 $make_e3
H &
        wttpccd1 $LOAD4_D $make_s4 $make_e4
H &
        wttpccd1 $LOAD5_D $make_s5 $make_e5
H &
        wttpccd1 $LOAD6_D $make_s6 $make_e6
H &
        wttpccd1 $LOAD7_D $make_s7 $make_e7
H &
        wttpccd1 $LOAD8_D $make_s8 $make_e8
H &

        wait

foreach num ( 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
31 )

        @ dsi_num1 = $num * 8 + 1
        @ dsi_num2 = $num * 8 + 2
        @ dsi_num3 = $num * 8 + 3
        @ dsi_num4 = $num * 8 + 4
        @ dsi_num5 = $num * 8 + 5
        @ dsi_num6 = $num * 8 + 6
        @ dsi_num7 = $num * 8 + 7
        @ dsi_num8 = $num * 8 + 8

        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num1\_DSI \
                                        -s
$WK1_D \
                                        -s
$WK2_D \
                                        -
n $LOAD1_D/HI$make_s1\_$make_e1 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num2\_DSI \
                                        -s
$WK3_D \
                                        -s
$WK4_D \
                                        -
n $LOAD2_D/HI$make_s2\_$make_e2 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num3\_DSI \
                                        -s
$WK5_D \
                                        -s
$WK6_D \
                                        -
n $LOAD3_D/HI$make_s3\_$make_e3 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num4\_DSI \
                                        -s
$WK7_D \
                                        -s
$WK8_D \
                                        -
n $LOAD4_D/HI$make_s4\_$make_e4 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num5\_DSI \
                                        -s
$WK9_D \
                                        -s
$WK10_D \
                                        -
n $LOAD5_D/HI$make_s5\_$make_e5 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num6\_DSI \
                                        -s
$WK11_D \
                                        -s
$WK12_D \
                                        -
n $LOAD6_D/HI$make_s6\_$make_e6 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num7\_DSI \
                                        -s
$WK13_D \
                                        -s
$WK14_D \
                                        -
n $LOAD7_D/HI$make_s7\_$make_e7 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num8\_DSI \
                                        -s
$WK15_D \
                                        -s
$WK16_D \
                                        -
n $LOAD8_D/HI$make_s8\_$make_e8 &

        @ rm_make_s1 = $make_s1
        @ rm_make_e1 = $make_e1
        @ rm_make_s2 = $make_s2
        @ rm_make_e2 = $make_e2
        @ rm_make_s3 = $make_s3
        @ rm_make_e3 = $make_e3
        @ rm_make_s4 = $make_s4
        @ rm_make_e4 = $make_e4
        @ rm_make_s5 = $make_s5
        @ rm_make_e5 = $make_e5
        @ rm_make_s6 = $make_s6
        @ rm_make_e6 = $make_e6
        @ rm_make_s7 = $make_s7
        @ rm_make_e7 = $make_e7
        @ rm_make_s8 = $make_s8
        @ rm_make_e8 = $make_e8

        @ make_s1 = $num * 88 + 89
        @ make_e1 = $num * 88 + 99
        @ make_s2 = $num * 88 + 100
        @ make_e2 = $num * 88 + 110
        @ make_s3 = $num * 88 + 111
        @ make_e3 = $num * 88 + 121
        @ make_s4 = $num * 88 + 122
        @ make_e4 = $num * 88 + 132
        @ make_s5 = $num * 88 + 133
        @ make_e5 = $num * 88 + 143
        @ make_s6 = $num * 88 + 144
        @ make_e6 = $num * 88 + 154
        @ make_s7 = $num * 88 + 155
        @ make_e7 = $num * 88 + 165
        @ make_s8 = $num * 88 + 166
        @ make_e8 = $num * 88 + 176

        wttpccd1 $LOAD1_D $make_s1 $make_e1
H &
        wttpccd1 $LOAD2_D $make_s2 $make_e2
H &
```

```
        wttpccd1 $LOAD3_D $make_s3 $make_e3
H &
        wttpccd1 $LOAD4_D $make_s4 $make_e4
H &
        wttpccd1 $LOAD5_D $make_s5 $make_e5
H &
        wttpccd1 $LOAD6_D $make_s6 $make_e6
H &
        wttpccd1 $LOAD7_D $make_s7 $make_e7
H &
        wttpccd1 $LOAD8_D $make_s8 $make_e8
H &

        wait

    rm
$LOAD1_D/HI$rm_make_s1\_$rm_make_e1
    rm
$LOAD2_D/HI$rm_make_s2\_$rm_make_e2
    rm
$LOAD3_D/HI$rm_make_s3\_$rm_make_e3
    rm
$LOAD4_D/HI$rm_make_s4\_$rm_make_e4
    rm
$LOAD5_D/HI$rm_make_s5\_$rm_make_e5
    rm
$LOAD6_D/HI$rm_make_s6\_$rm_make_e6
    rm
$LOAD7_D/HI$rm_make_s7\_$rm_make_e7
    rm
$LOAD8_D/HI$rm_make_s8\_$rm_make_e8

end

    rm $LOAD1_D/HI$make_s1\_$make_e1
    rm $LOAD2_D/HI$make_s2\_$make_e2
    rm $LOAD3_D/HI$make_s3\_$make_e3
    rm $LOAD4_D/HI$make_s4\_$make_e4
    rm $LOAD5_D/HI$make_s5\_$make_e5
    rm $LOAD6_D/HI$make_s6\_$make_e6
    rm $LOAD7_D/HI$make_s7\_$make_e7
    rm $LOAD8_D/HI$make_s8\_$make_e8


## STOCK


        @ make_s1 = 1
        @ make_e1 = 16
        @ make_s11 = 17
        @ make_e11 = 33
        @ make_s2 = 34
        @ make_e2 = 49
        @ make_s12 = 50
        @ make_e12 = 66
        @ make_s3 = 67
        @ make_e3 = 82
        @ make_s13 = 83
        @ make_e13 = 99
        @ make_s4 = 100
        @ make_e4 = 115
        @ make_s14 = 116
        @ make_e14 = 132
        @ make_s5 = 133
        @ make_e5 = 148
        @ make_s15 = 149
        @ make_e15 = 165
        @ make_s6 = 166
        @ make_e6 = 181
        @ make_s16 = 182
        @ make_e16 = 198
        @ make_s7 = 199
        @ make_e7 = 214
        @ make_s17 = 215
        @ make_e17 = 231
        @ make_s8 = 232
        @ make_e8 = 247
        @ make_s18 = 248
        @ make_e18 = 264

        wttpccd1 $LOAD1_D $make_s1 $make_e1
S &
        wttpccd1 $LOAD2_D $make_s2 $make_e2
S &
        wttpccd1 $LOAD3_D $make_s3 $make_e3
S &
        wttpccd1 $LOAD4_D $make_s4 $make_e4
S &
        wttpccd1 $LOAD5_D $make_s5 $make_e5
S &
        wttpccd1 $LOAD6_D $make_s6 $make_e6
S &
        wttpccd1 $LOAD7_D $make_s7 $make_e7
S &
        wttpccd1 $LOAD8_D $make_s8 $make_e8
S &
        wttpccd1 $LOAD9_D $make_s11
$make_e11 S &
        wttpccd1 $LOAD10_D $make_s12
$make_e12 S &
        wttpccd1 $LOAD11_D $make_s13
$make_e13 S &
        wttpccd1 $LOAD12_D $make_s14
$make_e14 S &
        wttpccd1 $LOAD13_D $make_s15
$make_e15 S &
        wttpccd1 $LOAD14_D $make_s16
$make_e16 S &
        wttpccd1 $LOAD15_D $make_s17
$make_e17 S &
        wttpccd1 $LOAD16_D $make_s18
$make_e18 S &

        wait

foreach num ( 0 1 2 3 4 5 6 7 8 9 10 )

        @ dsi_num1 = $num * 8 + 1
        @ dsi_num2 = $num * 8 + 2
        @ dsi_num3 = $num * 8 + 3
        @ dsi_num4 = $num * 8 + 4
        @ dsi_num5 = $num * 8 + 5
        @ dsi_num6 = $num * 8 + 6
        @ dsi_num7 = $num * 8 + 7
        @ dsi_num8 = $num * 8 + 8

        timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num1\_DSI \
                                        -s
$WK1_D \
                                        -s
$WK2_D \
                                        -
n $LOAD1_D/ST$make_s1\_$make_e1 \

$LOAD9_D/ST$make_s11\_$make_e11 &

        timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num2\_DSI \
                                        -s
$WK3_D \
                                        -s
$WK4_D \
                                        -
n $LOAD2_D/ST$make_s2\_$make_e2 \

$LOAD10_D/ST$make_s12\_$make_e12 &


        timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num7\_DSI \
                                        -s
$WK13_D \
                                        -s
$WK14_D \
                                        -
n $LOAD7_D/ST$make_s7\_$make_e7 \

$LOAD15_D/ST$make_s17\_$make_e17 &

        timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num8\_DSI \
                                        -s
$WK15_D \
                                        -s
$WK16_D \
                                        -
n $LOAD8_D/ST$make_s8\_$make_e8 \

$LOAD16_D/ST$make_s18\_$make_e18 &

        timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num5\_DSI \
                                        -s
$WK9_D \
                                        -s
$WK10_D \
                                        -
n $LOAD5_D/ST$make_s5\_$make_e5 \

$LOAD13_D/ST$make_s15\_$make_e15 &

        timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num6\_DSI \
                                        -s
$WK11_D \
                                        -s
$WK12_D \
                                        -
n $LOAD6_D/ST$make_s6\_$make_e6 \

$LOAD14_D/ST$make_s16\_$make_e16 &

        timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num3\_DSI \
                                        -s
$WK5_D \
                                        -s
$WK6_D \
                                        -
n $LOAD3_D/ST$make_s3\_$make_e3 \

$LOAD11_D/ST$make_s13\_$make_e13 &
```

```
        timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num4\_DSI \
                                        -s
$WK7_D \
                                        -s
$WK8_D \
                                        -
n $LOAD4_D/ST$make_s4\_$make_e4 \

$LOAD12_D/ST$make_s14\_$make_e14 &

        @ rm_make_s1 = $make_s1
        @ rm_make_e1 = $make_e1
        @ rm_make_s11 = $make_s11
        @ rm_make_e11 = $make_e11
        @ rm_make_s2 = $make_s2
        @ rm_make_e2 = $make_e2
        @ rm_make_s12 = $make_s12
        @ rm_make_e12 = $make_e12
        @ rm_make_s3 = $make_s3
        @ rm_make_e3 = $make_e3
        @ rm_make_s13 = $make_s13
        @ rm_make_e13 = $make_e13
        @ rm_make_s4 = $make_s4
        @ rm_make_e4 = $make_e4
        @ rm_make_s14 = $make_s14
        @ rm_make_e14 = $make_e14
        @ rm_make_s5 = $make_s5
        @ rm_make_e5 = $make_e5
        @ rm_make_s15 = $make_s15
        @ rm_make_e15 = $make_e15
        @ rm_make_s6 = $make_s6
        @ rm_make_e6 = $make_e6
        @ rm_make_s16 = $make_s16
        @ rm_make_e16 = $make_e16
        @ rm_make_s7 = $make_s7
        @ rm_make_e7 = $make_e7
        @ rm_make_s17 = $make_s17
        @ rm_make_e17 = $make_e17
        @ rm_make_s8 = $make_s8
        @ rm_make_e8 = $make_e8
        @ rm_make_s18 = $make_s18
        @ rm_make_e18 = $make_e18


        @ make_s1 = $num * 264 + 265
        @ make_e1 = $num * 264 + 280
        @ make_s11 = $num * 264 + 281
        @ make_e11 = $num * 264 + 297
        @ make_s2 = $num * 264 + 298
        @ make_e2 = $num * 264 + 313
        @ make_s12 = $num * 264 + 314
        @ make_e12 = $num * 264 + 330
        @ make_s3 = $num * 264 + 331
        @ make_e3 = $num * 264 + 346
        @ make_s13 = $num * 264 + 347
        @ make_e13 = $num * 264 + 363
        @ make_s4 = $num * 264 + 364
        @ make_e4 = $num * 264 + 379
        @ make_s14 = $num * 264 + 380
        @ make_e14 = $num * 264 + 396
        @ make_s5 = $num * 264 + 397
        @ make_e5 = $num * 264 + 412
        @ make_s15 = $num * 264 + 413
        @ make_e15 = $num * 264 + 429
        @ make_s6 = $num * 264 + 430
        @ make_e6 = $num * 264 + 445
        @ make_s16 = $num * 264 + 446
        @ make_e16 = $num * 264 + 462
        @ make_s7 = $num * 264 + 463
        @ make_e7 = $num * 264 + 478
        @ make_s17 = $num * 264 + 479
        @ make_e17 = $num * 264 + 495
        @ make_s8 = $num * 264 + 496
        @ make_e8 = $num * 264 + 511
        @ make_s18 = $num * 264 + 512
        @ make_e18 = $num * 264 + 528

        wttpccd1 $LOAD1_D $make_s1 $make_e1
S &
        wttpccd1 $LOAD2_D $make_s2 $make_e2
S &
        wttpccd1 $LOAD3_D $make_s3 $make_e3
S &
        wttpccd1 $LOAD4_D $make_s4 $make_e4
S &
        wttpccd1 $LOAD5_D $make_s5 $make_e5
S &
        wttpccd1 $LOAD6_D $make_s6 $make_e6
S &
        wttpccd1 $LOAD7_D $make_s7 $make_e7
S &
        wttpccd1 $LOAD8_D $make_s8 $make_e8
S &
        wttpccd1 $LOAD9_D $make_s11
$make_e11 S &
        wttpccd1 $LOAD10_D $make_s12
$make_e12 S &
        wttpccd1 $LOAD11_D $make_s13
$make_e13 S &
        wttpccd1 $LOAD12_D $make_s14
$make_e14 S &
        wttpccd1 $LOAD13_D $make_s15
$make_e15 S &
        wttpccd1 $LOAD14_D $make_s16
$make_e16 S &
        wttpccd1 $LOAD15_D $make_s17
$make_e17 S &
        wttpccd1 $LOAD16_D $make_s18
$make_e18 S &

        wait

        rm
$LOAD1_D/ST$rm_make_s1\_$rm_make_e1 &
        rm
$LOAD9_D/ST$rm_make_s11\_$rm_make_e11 &
        rm
$LOAD2_D/ST$rm_make_s2\_$rm_make_e2 &
        rm
$LOAD10_D/ST$rm_make_s12\_$rm_make_e12
&
        rm
$LOAD7_D/ST$rm_make_s7\_$rm_make_e7 &
        rm
$LOAD15_D/ST$rm_make_s17\_$rm_make_e17
&
        rm
$LOAD8_D/ST$rm_make_s8\_$rm_make_e8 &
        rm
$LOAD16_D/ST$rm_make_s18\_$rm_make_e18
&
        rm
$LOAD5_D/ST$rm_make_s5\_$rm_make_e5 &
        rm
$LOAD13_D/ST$rm_make_s15\_$rm_make_e15
&
        rm
$LOAD6_D/ST$rm_make_s6\_$rm_make_e6 &
        rm
$LOAD14_D/ST$rm_make_s16\_$rm_make_e16
&
        rm
$LOAD3_D/ST$rm_make_s3\_$rm_make_e3 &
        rm
$LOAD11_D/ST$rm_make_s13\_$rm_make_e13
&
        rm
$LOAD4_D/ST$rm_make_s4\_$rm_make_e4 &
        rm
$LOAD12_D/ST$rm_make_s14\_$rm_make_e14
&

end
        rm $LOAD1_D/ST$make_s1\_$make_e1 &
        rm $LOAD9_D/ST$make_s11\_$make_e11
&
        rm $LOAD2_D/ST$make_s2\_$make_e2 &
        rm
$LOAD10_D/ST$make_s12\_$make_e12 &
        rm $LOAD7_D/ST$make_s7\_$make_e7 &
        rm
$LOAD15_D/ST$make_s17\_$make_e17 &
        rm $LOAD8_D/ST$make_s8\_$make_e8 &
        rm
$LOAD16_D/ST$make_s18\_$make_e18 &
        rm $LOAD5_D/ST$make_s5\_$make_e5 &
        rm
$LOAD13_D/ST$make_s15\_$make_e15 &
        rm $LOAD6_D/ST$make_s6\_$make_e6 &
        rm
$LOAD14_D/ST$make_s16\_$make_e16 &
        rm $LOAD3_D/ST$make_s3\_$make_e3 &
        rm
$LOAD11_D/ST$make_s13\_$make_e13 &
        rm $LOAD4_D/ST$make_s4\_$make_e4 &
        rm
$LOAD12_D/ST$make_s14\_$make_e14 &


## ORDERS ORDERLINE NEWORDER

        @ make_s1 = 1
        @ make_e1 = 11
        @ make_s2 = 12
        @ make_e2 = 22
        @ make_s3 = 23
        @ make_e3 = 33
        @ make_s4 = 34
        @ make_e4 = 44
        @ make_s5 = 45
        @ make_e5 = 55
        @ make_s6 = 56
        @ make_e6 = 66
        @ make_s7 = 67
        @ make_e7 = 77
        @ make_s8 = 78
        @ make_e8 = 88

        wttpccd1 $LOAD1_D $make_s1 $make_e1
O &
```

```
        wttpccd1 $LOAD2_D $make_s2 $make_e2
O &
        wttpccd1 $LOAD3_D $make_s3 $make_e3
O &
        wttpccd1 $LOAD4_D $make_s4 $make_e4
O &
        wttpccd1 $LOAD5_D $make_s5 $make_e5
O &
        wttpccd1 $LOAD6_D $make_s6 $make_e6
O &
        wttpccd1 $LOAD7_D $make_s7 $make_e7
O &
        wttpccd1 $LOAD8_D $make_s8 $make_e8
O &

        wait


foreach num ( 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
31 )

        @ dsi_num_os1 = $num * 8 * 1 + 1
        @ dsi_num_os2 = $num * 8 * 1 + 2
        @ dsi_num_os3 = $num * 8 * 1 + 3
        @ dsi_num_os4 = $num * 8 * 1 + 4
        @ dsi_num_os5 = $num * 8 * 1 + 5
        @ dsi_num_os6 = $num * 8 * 1 + 6
        @ dsi_num_os7 = $num * 8 * 1 + 7
        @ dsi_num_os8 = $num * 8 * 1 + 8
        @ dsi_num_ol1 = $num * 8 * 1 + 1
        @ dsi_num_ol2 = $num * 8 * 1 + 2
        @ dsi_num_ol3 = $num * 8 * 1 + 3
        @ dsi_num_ol4 = $num * 8 * 1 + 4
        @ dsi_num_ol5 = $num * 8 * 1 + 5
        @ dsi_num_ol6 = $num * 8 * 1 + 6
        @ dsi_num_ol7 = $num * 8 * 1 + 7
        @ dsi_num_ol8 = $num * 8 * 1 + 8
        @ dsi_num_no1 = $num * 8 * 1 + 1
        @ dsi_num_no2 = $num * 8 * 1 + 2
        @ dsi_num_no3 = $num * 8 * 1 + 3
        @ dsi_num_no4 = $num * 8 * 1 + 4
        @ dsi_num_no5 = $num * 8 * 1 + 5
        @ dsi_num_no6 = $num * 8 * 1 + 6
        @ dsi_num_no7 = $num * 8 * 1 + 7
        @ dsi_num_no8 = $num * 8 * 1 + 8

        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os1\_DSI -h -f 10\
                                                -s
$WK1_D \
                                                -s
$WK2_D \

        $LOAD1_D/OS$make_s1\_$make_e1 &
wait
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os2\_DSI -h -f 10\
                                                -s
$WK3_D \
                                                -s
$WK4_D \

        $LOAD2_D/OS$make_s2\_$make_e2 &
wait
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os3\_DSI -h -f 10\
                                                -s
$WK5_D \
                                                -s
$WK6_D \

        $LOAD3_D/OS$make_s3\_$make_e3 &
wait
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os4\_DSI -h -f 10\
                                                -s
$WK7_D \
                                                -s
$WK8_D \

        $LOAD4_D/OS$make_s4\_$make_e4 &
wait
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os5\_DSI -h -f 10\
                                                -s
$WK9_D \
                                                -s
$WK10_D \

        $LOAD5_D/OS$make_s5\_$make_e5 &
wait
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os6\_DSI -h -f 10\
                                                -s
$WK11_D \
                                                -s
$WK12_D \

        $LOAD6_D/OS$make_s6\_$make_e6 &
wait
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os7\_DSI -h -f 10\
                                                -s
$WK13_D \
                                                -s
$WK14_D \

        $LOAD7_D/OS$make_s7\_$make_e7 &
wait
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os8\_DSI -h -f 10\
                                                -s
$WK15_D \
                                                -s
$WK16_D \

        $LOAD8_D/OS$make_s8\_$make_e8 &

        wait
        timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no1\_DSI -h -f
20 \
                                                -s
$WK1_D \
                                                -s
$WK2_D \
                                                -
n $LOAD1_D/NO$make_s1\_$make_e1 &
wait
        timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no2\_DSI -h -f
20 \
                                                -s
$WK3_D \
                                                -s
$WK4_D \
                                                -
n $LOAD2_D/NO$make_s2\_$make_e2 &
wait
        timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no3\_DSI -h -f
20 \
                                                -s
$WK5_D \
                                                -s
$WK6_D \
                                                -
n $LOAD3_D/NO$make_s3\_$make_e3 &
wait
        timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no4\_DSI -h -f
20 \
                                                -s
$WK7_D \
                                                -s
$WK8_D \
                                                -
n $LOAD4_D/NO$make_s4\_$make_e4 &
wait
        timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no5\_DSI -h -f
20 \
                                                -s
$WK9_D \
                                                -s
$WK10_D \
                                                -
n $LOAD5_D/NO$make_s5\_$make_e5 &
wait
        timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no6\_DSI -h -f
20 \
                                                -s
$WK11_D \
                                                -s
$WK12_D \
                                                -
n $LOAD6_D/NO$make_s6\_$make_e6 &
wait
        timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no7\_DSI -h -f
20 \
                                                -s
$WK13_D \
                                                -s
$WK14_D \
                                                -
n $LOAD7_D/NO$make_s7\_$make_e7 &
wait
        timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no8\_DSI -h -f
20 \
                                                -s
$WK15_D \
                                                -s
$WK16_D \
                                                -
n $LOAD8_D/NO$make_s8\_$make_e8 &
        wait

        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol1\_DSI -h \
```

```
                                        -s
$WK1_D \
                                        -s
$WK2_D \

        $LOAD1_D/OL$make_s1\_$make_e1 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol2\_DSI -h \
                                        -s
$WK3_D \
                                        -s
$WK4_D \

        $LOAD2_D/OL$make_s2\_$make_e2 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol3\_DSI -h \
                                        -s
$WK5_D \
                                        -s
$WK6_D \

        $LOAD3_D/OL$make_s3\_$make_e3 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol4\_DSI -h \
                                        -s
$WK7_D \
                                        -s
$WK8_D \

        $LOAD4_D/OL$make_s4\_$make_e4 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol5\_DSI -h \
                                        -s
$WK9_D \
                                        -s
$WK10_D \

        $LOAD5_D/OL$make_s5\_$make_e5 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol6\_DSI -h \
                                        -s
$WK11_D \
                                        -s
$WK12_D \

        $LOAD6_D/OL$make_s6\_$make_e6 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol7\_DSI -h \
                                        -s
$WK13_D \
                                        -s
$WK14_D \

        $LOAD7_D/OL$make_s7\_$make_e7 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol8\_DSI -h \
                                        -s
$WK15_D \
                                        -s
$WK16_D \

        $LOAD8_D/OL$make_s8\_$make_e8 &


        @ rm_make_s1 = $make_s1
        @ rm_make_e1 = $make_e1
        @ rm_make_s2 = $make_s2
        @ rm_make_e2 = $make_e2
        @ rm_make_s3 = $make_s3
        @ rm_make_e3 = $make_e3
        @ rm_make_s4 = $make_s4
        @ rm_make_e4 = $make_e4
        @ rm_make_s5 = $make_s5
        @ rm_make_e5 = $make_e5
        @ rm_make_s6 = $make_s6
        @ rm_make_e6 = $make_e6
        @ rm_make_s7 = $make_s7
        @ rm_make_e7 = $make_e7
        @ rm_make_s8 = $make_s8
        @ rm_make_e8 = $make_e8

        @ make_s1 = $num * 88 + 89
        @ make_e1 = $num * 88 + 99
        @ make_s2 = $num * 88 + 100
        @ make_e2 = $num * 88 + 110
        @ make_s3 = $num * 88 + 111
        @ make_e3 = $num * 88 + 121
        @ make_s4 = $num * 88 + 122
        @ make_e4 = $num * 88 + 132
        @ make_s5 = $num * 88 + 133
        @ make_e5 = $num * 88 + 143
        @ make_s6 = $num * 88 + 144
        @ make_e6 = $num * 88 + 154
        @ make_s7 = $num * 88 + 155
        @ make_e7 = $num * 88 + 165
        @ make_s8 = $num * 88 + 166
        @ make_e8 = $num * 88 + 176

        wttpccd1 $LOAD1_D $make_s1 $make_e1
O &
        wttpccd1 $LOAD2_D $make_s2 $make_e2
O &
        wttpccd1 $LOAD3_D $make_s3 $make_e3
O &
        wttpccd1 $LOAD4_D $make_s4 $make_e4
O &
        wttpccd1 $LOAD5_D $make_s5 $make_e5
O &
        wttpccd1 $LOAD6_D $make_s6 $make_e6
O &
        wttpccd1 $LOAD7_D $make_s7 $make_e7
O &
        wttpccd1 $LOAD8_D $make_s8 $make_e8
O &

        wait


        rm
$LOAD1_D/OS$rm_make_s1\_$rm_make_e1 &
        rm
$LOAD2_D/OS$rm_make_s2\_$rm_make_e2 &
        rm
$LOAD3_D/OS$rm_make_s3\_$rm_make_e3 &
        rm
$LOAD4_D/OS$rm_make_s4\_$rm_make_e4 &
        rm
$LOAD5_D/OS$rm_make_s5\_$rm_make_e5 &
        rm
$LOAD6_D/OS$rm_make_s6\_$rm_make_e6 &
        rm
$LOAD7_D/OS$rm_make_s7\_$rm_make_e7 &
        rm
$LOAD8_D/OS$rm_make_s8\_$rm_make_e8 &


        rm
$LOAD1_D/NO$rm_make_s1\_$rm_make_e1 &
        rm
$LOAD2_D/NO$rm_make_s2\_$rm_make_e2 &
        rm
$LOAD3_D/NO$rm_make_s3\_$rm_make_e3 &
        rm
$LOAD4_D/NO$rm_make_s4\_$rm_make_e4 &
        rm
$LOAD5_D/NO$rm_make_s5\_$rm_make_e5 &
        rm
$LOAD6_D/NO$rm_make_s6\_$rm_make_e6 &
        rm
$LOAD7_D/NO$rm_make_s7\_$rm_make_e7 &
        rm
$LOAD8_D/NO$rm_make_s8\_$rm_make_e8 &


        rm
$LOAD1_D/OL$rm_make_s1\_$rm_make_e1 &
        rm
$LOAD2_D/OL$rm_make_s2\_$rm_make_e2 &
        rm
$LOAD3_D/OL$rm_make_s3\_$rm_make_e3 &
        rm
$LOAD4_D/OL$rm_make_s4\_$rm_make_e4 &
        rm
$LOAD5_D/OL$rm_make_s5\_$rm_make_e5 &
        rm
$LOAD6_D/OL$rm_make_s6\_$rm_make_e6 &
        rm
$LOAD7_D/OL$rm_make_s7\_$rm_make_e7 &
        rm
$LOAD8_D/OL$rm_make_s8\_$rm_make_e8 &

end


        rm $LOAD1_D/OS$make_s1\_$make_e1 &
        rm $LOAD2_D/OS$make_s2\_$make_e2 &
        rm $LOAD3_D/OS$make_s3\_$make_e3 &
        rm $LOAD4_D/OS$make_s4\_$make_e4 &
        rm $LOAD5_D/OS$make_s5\_$make_e5 &
        rm $LOAD6_D/OS$make_s6\_$make_e6 &
        rm $LOAD7_D/OS$make_s7\_$make_e7 &
        rm $LOAD8_D/OS$make_s8\_$make_e8 &

        rm $LOAD1_D/NO$make_s1\_$make_e1 &
        rm $LOAD2_D/NO$make_s2\_$make_e2 &
        rm $LOAD3_D/NO$make_s3\_$make_e3 &
        rm $LOAD4_D/NO$make_s4\_$make_e4 &
        rm $LOAD5_D/NO$make_s5\_$make_e5 &
        rm $LOAD6_D/NO$make_s6\_$make_e6 &
        rm $LOAD7_D/NO$make_s7\_$make_e7 &
        rm $LOAD8_D/NO$make_s8\_$make_e8 &

        rm $LOAD1_D/OL$make_s1\_$make_e1 &
        rm $LOAD2_D/OL$make_s2\_$make_e2 &
        rm $LOAD3_D/OL$make_s3\_$make_e3 &
        rm $LOAD4_D/OL$make_s4\_$make_e4 &
        rm $LOAD5_D/OL$make_s5\_$make_e5 &
        rm $LOAD6_D/OL$make_s6\_$make_e6 &
        rm $LOAD7_D/OL$make_s7\_$make_e7 &
        rm $LOAD8_D/OL$make_s8\_$make_e8 &

wait
```

File: mkarc.sh

```
:
set -x
date

timex rdblog -G -a /DEV/rdsk/ARC1 2035M
sleep 1

timex rdblog -U -a /DEV/rdsk/ARC2
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC3
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC4
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC5
sleep 1

#exit

timex rdblog -U -a /DEV/rdsk/ARC6
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC7
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC8
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC9
sleep 1
timex rdblog -U -a /DEV/rdsk/ARC10
sleep 1
date
```

File: mktmplog.sh

```
:
set -x
date
echo "its takes 25 min (990424 9GB not duplex)"
rm /rdbptc/tpcc80/SYS/rdblogmanage
timex rdblog -l

LOG_AI=/DEV/rdsk/AI_LOG
LOG_BI=/DEV/rdsk/BI_LOG
LOG_IX=/DEV/rdsk/IX_LOG

rm /rdbptc/arc/ARC* &

timex rdblog -G -t -c 6600M -io 2048 $LOG_IX
$LOG_BI $LOG_AI 2040M 8000M 510


wait
```

File: sh.rdbups

```
#!/usr/bin/csh

rdbups -i TPCC.WAREHOUSE_1_DSI &
rdbups -i TPCC.DISTRICT_1_DSI &
rdbups -i TPCC.ITEM_1_DSI &
rdbups -i TPCC.CUSTOMER_1_DSI &
rdbups -i TPCC.CUSTOMER_X_1_DSI &


rdbups -i TPCC.ORDERS_1_DSI &
rdbups -i TPCC.ORDERS_IX_1_DSI &
rdbups -i TPCC.NEWORDER_1_DSI &
rdbups -i TPCC.NEWORDER_X_1_DSI &
rdbups -i TPCC.HISTORY_1_DSI &


rdbups -i TPCC.ORDERLIN_1_DSI &

rdbups -i TPCC.STOCK_1_DSI &

wait
```

File: wttpccd1.c

```
/*
 *      File Name   : wttpccd.ec
 *      Function Name     : main()
 *                : item()
 *                : warehouse()
 *                : stock()
 *                : district()
 *                : customer()
 *                : history()
 *                : orders()
 *                                  : new_order()
 *                                  : make_address()
 *                                  : lastname()
 *                : make_alpha_string()
 *                : make_number_string()
 *                : random_number()
 *                : set_seed()
 *                : nurand()
 *                                  : init_permutation()
 *      Description : DB  tpcc              , item,
        warehouse, stock,
 *                        district, customer, history,
orders, order_line,
 *                        new_order       ,
 *                                  (char )       .
 *      Author                :
 *      Reviewer    :
 *
 *      COPYRIGHT FUJITSU Limited 1995
 *
 *              95-03-13
 *                        -                         (  : %20s
==> %-s)
 *                        -
 *                         sprintf & fwrite    fprintf
(     )
 *                        - ORDERS  ORDER_LINE
NULL
 *
 *              95-05-16
 *                        - 10warehouse
 *              96-04-18
 *                        - W-TAX, D-TAX, C-DISCOUNT,
I-PRICE, OL-AMOUNT, H-AMOUNT
 *                                  (DECIMAL ->
SMALLINT or INTEGER)
 *                        - C-SINCE, OL-DELIVERY-D,
O-ENTRY-D
 *                                  (DECIMAL ->
CHAR)
 *              96-09-06
 *                        - file
 *                                  1.option          table
(3      parameter)
 *                                   (0..all, 1..IT/ST/HI/CU,
2..WH/DI/OL/OS/NO)
 *                                  2.file
 *
(/rdb/loaddata/[table  ]/[warehouse       _    ]
 *                                   ex.:
/rdb/loaddata/Customer/10_15 .. Customer Wh10-
15)
 *                        - Text      Binary      . DECIMAL
 *                                             :)
 *                         <decimal(a,b)  format>
 *                                  decimal(10,2)
aa|aa|aa|aa|ab|bs (6byte=a/2+1)
 *                                  a=decimal
(b              )
 *                                  b=              (s
8bit       )
 *                                  s=      . (+)    "c", (-)
"d"
 *                                  ex.) +12345678.23
= '(0x) 01|23|45|67|82|3c
 *                                 program          FUNC.
 *
 *                               "record    "
(
 *                                      )
 *                                  ex.) w_ytd = -
123.45;
 *                                                   :
        :
 *                                                   :
        :
 *                                          ("record
"    )
 *                                           w_ytd_1 =
0x00;
 *                                           w_ytd_2 =
0x00;
 *                                           w_ytd_3 =
0x00;
 *                                           w_ytd_4 =
0x00;
 *                                           w_ytd_5 =
0x12;
 *                                           w_ytd_6 =
0x34;
 *                                           w_ytd_7 =
0x5d;
 *
 *              96-11-27
 *                        - ol_i_id          (?)
 *                                  ol_i_id    1  10      n
 *                                  (  : n=2;
2,4,6,....99998,100000)
 *                                   n    setenv
TPCRANDBY n
 *                                  n < TPCRANDBY < 10
or undefined      == 1
```

```
 *
 *              97-02-18
 *                      - C_ID, H_C_ID, O_C_ID
 *                                      (SMALLINT ->
INTEGER)
 *                      - I_IM_ID
 *
 *              97-02-18
 *                      - fprintf -> sprintf + fwrite
 *
( item,stock,customer,history,orders,orderline )
 *                      - random_number  mac
 *                      - make_alpha_string
make_number_string
 *              rund        .(                  )
 *                      -
make_alpha_string                       .
 *              ( ORACLE , HP           )
 *                      - get_permutation       ,o_c_id
 *                      - TAB ID 221(c_last NURand
C)
 *              Issue       : C Value For NURand
 *              Specification : TPC-C,Clause 2.1.6
 *              c_last  NURand     C                .
 *               C-Load : DB                    C
 *               C-Run  :     (tran)                        C
 *               C-Delta: | C-Load - C-Run |
 *                  C        ,[ 0,255]
 *               C-Delta       ,[65,119]     ,96,112
 *              C-Run  , 111              .
 *                  Online : tranmain  Const
 *                  Online : pptpcc2(shell)  u14i
 *              nurand()
 *                  97-02-24
 *                      - fopen + fwrite -> open + write
 *
( item,stock,customer,history,orders,orderline )
 *                  97-02-25
 *                      - CUSTOMER,HISTORY
 *                  97-02-27
 *                      - sprintf    (typedef struct)
 *           - c_phone[16] -> c_phone[17]
 *                  97-03-05
 *                      - c_payment_cnt = 1 -> 0
 *                  97-03-11
 *                      -
make_alpha_string                       ,             .
 *              ( ORACLE     )
 *                  98-11-06
 *                      -                   100
 *                        w_ytd, d_ytd, c_credit_lim,
c_balance, c_ytd_payment
*/

#include <stdio.h>
#include <string.h>
#include <sys/types.h>  /* 1994.12.28 add
kawabata */
#include <sys/stat.h>     /* 1995.02.24 add
arakawa */
#include <fcntl.h>                  /* 1995.02.24 add
arakawa */
#include <time.h>
#include <stdlib.h>                 /* 1996.11.27
added K.Fukui for "getenv()" */
#include <unistd.h>     /* 1997.02.24 write
*/

/*#define     DBNAME                  "tpcc"*/      /*
DB   */
#define MAXITEMS              100000            /*
*/
#define MAXSTOCK              100000            /*
STOCK     */
#define DIST_PER_WARE         10          /*
*/
#define CUST_PER_DIST         3000        /*
*/
#define ORD_PER_DIST          3000        /*
*/
#define NEWORDS                       900         /*
*/
#define CLS_CNT                       10000
        /*                      */

#define CMT_CNT                               3

#define       T256                16777216
#define       D256                65536
#define       NNUL_V                    0x00
#define       NUL_V                     0xFF

    /* 1997-02-27 sprintf    (typedef struct) */
typedef struct
  {
    char      i_id_1,i_id_2,i_id_3,i_id_4 ;
    char      i_im_id_1, i_im_id_2, i_im_id_3,
i_im_id_4 ;
    char   i_name[24] ;
    char      i_price_1, i_price_2 ;
    char   i_data[50] ;
  } item_str ;

typedef struct
  {
    char      d_id_1,d_id_2;
    char   d_w_id_1,d_w_id_2;
    char      d_name[10];
    char      d_street_1[20];
    char      d_street_2[20];
    char      d_city[20];
    char      d_state[2];
    char      d_zip[9];
    char      d_tax_1, d_tax_2;
    char      d_ytd_1, d_ytd_2, d_ytd_3, d_ytd_4,
d_ytd_5, d_ytd_6, d_ytd_7;
    char      d_next_o_id_1, d_next_o_id_2,
d_next_o_id_3, d_next_o_id_4;
  } district_str ;

typedef struct
  {
    char      w_id_1, w_id_2;
    char      w_name[10];
    char      w_street_1[20];
    char      w_street_2[20];
    char      w_city[20];
    char      w_state[2];
    char      w_zip[9];
    char      w_tax_1, w_tax_2;
    char      w_ytd_1, w_ytd_2, w_ytd_3, w_ytd_4,
w_ytd_5, w_ytd_6, w_ytd_7;
  } warehouse_str ;

typedef struct
  {
    char      s_i_id_1, s_i_id_2, s_i_id_3, s_i_id_4;
    char      s_w_id_1, s_w_id_2;
    char   s_quantity_1, s_quantity_2;
    char      s_dist_01[24];
    char      s_dist_02[24];
    char      s_dist_03[24];
    char      s_dist_04[24];
    char      s_dist_05[24];
    char      s_dist_06[24];
    char      s_dist_07[24];
    char      s_dist_08[24];
    char      s_dist_09[24];
    char      s_dist_10[24];
    char   s_ytd_1, s_ytd_2, s_ytd_3, s_ytd_4;
    char      s_order_cnt_1, s_order_cnt_2;
    char   s_remote_cnt_1, s_remote_cnt_2;
    char      s_data[50];
  } stock_str ;

typedef struct
  {
    char   c_id_1, c_id_2, c_id_3, c_id_4;
    char      c_d_id_1, c_d_id_2;
    char      c_w_id_1, c_w_id_2;
    char      c_first[16];
    char      c_middle[2];
    char      c_last[16];
    char      c_street_1[20];
    char      c_street_2[20];
    char      c_city[20];
    char      c_state[2];
    char      c_zip[9];
    char      c_phone[16];
    char      c_since[14];
    char      c_credit[2];
    char      c_credit_lim_1, c_credit_lim_2,
c_credit_lim_3, c_credit_lim_4;
    char   c_credit_lim_5, c_credit_lim_6,
c_credit_lim_7;
    char      c_discount_1, c_discount_2;
    char      c_balance_1, c_balance_2,
c_balance_3, c_balance_4;
    char      c_balance_5, c_balance_6,
c_balance_7;
    char   c_ytd_payment_1, c_ytd_payment_2,
c_ytd_payment_3, c_ytd_payment_4;
    char      c_ytd_payment_5, c_ytd_payment_6,
c_ytd_payment_7;
    char      c_payment_cnt_1, c_payment_cnt_2;
    char      c_delivery_cnt_1, c_delivery_cnt_2;
    char      c_data[500];
  } customer_str ;

typedef struct
  {
    char      h_c_id_1, h_c_id_2,h_c_id_3,
h_c_id_4;
    char      h_c_d_id_1, h_c_d_id_2;
    char      h_c_w_id_1, h_c_w_id_2;
    char      h_d_id_1, h_d_id_2;
    char      h_w_id_1, h_w_id_2;
    char      h_date[14];
    char      h_amount_1, h_amount_2,
h_amount_3, h_amount_4;
    char      h_data[24];
  } history_str ;

typedef struct
```

```
 {
   char   o_id_v1, o_id_v2;
    char      o_id_1, o_id_2, o_id_3, o_id_4;
   char   o_d_id_v1, o_d_id_v2;
    char      o_d_id_1, o_d_id_2;
   char   o_w_id_v1, o_w_id_v2;
    char      o_w_id_1, o_w_id_2;
   char   o_c_id_v1, o_c_id_v2;
    char      o_c_id_1, o_c_id_2, o_c_id_3,
o_c_id_4;
   char   o_entry_d_v1,o_entry_d_v2;
    char      o_entry_d[14];
   char   o_carrier_id_v1, o_carrier_id_v2;
    char      o_carrier_id_1, o_carrier_id_2;
   char   o_ol_cnt_v1, o_ol_cnt_v2;
    char      o_ol_cnt_1, o_ol_cnt_2;
   char   o_all_local_v1, o_all_local_v2;
    char      o_all_local_1, o_all_local_2;
 } orders_str ;

typedef struct
 {
   char   ol_o_id_v1, ol_o_id_v2;
    char   ol_o_id_1, ol_o_id_2, ol_o_id_3,
ol_o_id_4;
   char   ol_d_id_v1, ol_d_id_v2;
    char      ol_d_id_1, ol_d_id_2;
   char   ol_w_id_v1, ol_w_id_v2;
    char      ol_w_id_1, ol_w_id_2;
   char   ol_number_v1, ol_number_v2;
    char   ol_number_1, ol_number_2;
   char   ol_i_id_v1, ol_i_id_v2;
    char      ol_i_id_1, ol_i_id_2, ol_i_id_3,
ol_i_id_4;
   char   ol_supply_w_id_v1, ol_supply_w_id_v2;
    char      ol_supply_w_id_1, ol_supply_w_id_2;
   char   ol_delivery_d_v1,ol_delivery_d_v2;
    char      ol_delivery_d[14];
   char   ol_quantity_v1, ol_quantity_v2;
    char      ol_quantity_1, ol_quantity_2;
   char   ol_amount_v1, ol_amount_v2;
    char   ol_amount_1, ol_amount_2,
ol_amount_3, ol_amount_4;
   char   ol_dist_info_v1,ol_dist_info_v2;
    char      ol_dist_info[24];
 } orderline_str ;

typedef struct
 {
    char      no_o_id_1, no_o_id_2, no_o_id_3,
no_o_id_4;
    char      no_d_id_1, no_d_id_2;
    char      no_w_id_1, no_w_id_2;
 } neworder_str ;

int len_i ;
   /* 1997-02-18 TAB ID 221(c_last NURand     C)
*/
#define C_DELTA          87                  /* |
C_LOAD - C_RAN |       */
#define C_RUN            111            /* TRAN
NURand    C       */
#define C_LOAD     (C_DELTA+C_RUN)
/* DB LOAD NURand    C       */

       /* 1997-02-18  fprintf -> sprintf + fwrite
*/

#define ITEM_SIZE              sizeof(item_str)
/* ITEM              (84)*/
#define DISTRICT_SIZE          sizeof(district_str)
/* DISTRICT          (98)*/
#define WAREHOUSE_SIZE
       sizeof(warehouse_str) /* WAREHOUSE
(92)*/
#define STOCK_SIZE             sizeof(stock_str)
/* STOCK             (306)*/
#define CUSTOMER_SIZE
       sizeof(customer_str)  /* CUSTOMER
(672)*/
#define HISTORY_SIZEsizeof(history_str)   /*
HISTORY           (54)*/
#define ORDERS_SIZE    sizeof(orders_str)    /*
ORDERS            (32+16)*/
#define ORDERLINE_SIZE     sizeof(orderline_str)
/* ORDERLINE         (60+20)*/
#define NEWORDER_SIZE
       sizeof(neworder_str)  /* HISTORY
( 8)*/

#define ITEM_COUNT       1024              /*
ITEM                 */
#define STOCK_COUNT            1024
/* STOCK                */
#define CUSTOMER_COUNT    512
/* CUSTOMER             */
#define HISTORY_COUNT    1024
/* HISTORY              */
#define ORDERS_COUNT     1024
/* ORDERS               */
#define ORDERLINE_COUNT  2048
/* ORDERLINE            */

/*             */
/*               */
char   yyyymmddhhmmss[15];

/*                  . ..._1  ..._7   96-09-06
*/
/*       (INTEGER:_1  _4 / SMALLINT:_1  _2 /
DECIMAL:_1  _7) */

int     i_id;
int             i_id_1, i_id_2, i_id_3, i_id_4;
int     i_im_id;
                     /* 97-02-18      */
int             i_im_id_1, i_im_id_2, i_im_id_3,
i_im_id_4;   /* 97-02-18      */
char   i_name[25];
int     i_price;
int             i_price_1, i_price_2;
char   i_data[51];

short  w_id;
int             w_id_1, w_id_2;
char   w_name[11];
char   w_street_1[21];
char   w_street_2[21];
char   w_city[21];
char   w_state[3];
char   w_zip[10];
int     w_tax;
int             w_tax_1, w_tax_2;
float  w_ytd;
int             w_ytd_1, w_ytd_2, w_ytd_3, w_ytd_4,
w_ytd_5, w_ytd_6, w_ytd_7;

int     s_i_id;
int             s_i_id_1, s_i_id_2, s_i_id_3, s_i_id_4;
short  s_w_id;
int             s_w_id_1, s_w_id_2;
int     s_quantity;
int             s_quantity_1, s_quantity_2;
char   s_dist_01[25];
char   s_dist_02[25];
char   s_dist_03[25];
char   s_dist_04[25];
char   s_dist_05[25];
char   s_dist_06[25];
char   s_dist_07[25];
char   s_dist_08[25];
char   s_dist_09[25];
char   s_dist_10[25];
int             s_ytd;
int             s_ytd_1, s_ytd_2, s_ytd_3, s_ytd_4;
int     s_order_cnt;
int             s_order_cnt_1, s_order_cnt_2;
int     s_remote_cnt;
int             s_remote_cnt_1, s_remote_cnt_2;
char   s_data[51];

short  d_id;
int             d_id_1,d_id_2;
short  d_w_id;
int             d_w_id_1,d_w_id_2;
char   d_name[11];
char   d_street_1[21];
char   d_street_2[21];
char   d_city[21];
char   d_state[3];
char   d_zip[10];
int             d_tax;
int             d_tax_1, d_tax_2;
char   work[10];
float  d_ytd;
int             d_ytd_1, d_ytd_2, d_ytd_3, d_ytd_4,
d_ytd_5, d_ytd_6, d_ytd_7;
int     d_next_o_id;
int             d_next_o_id_1, d_next_o_id_2,
d_next_o_id_3, d_next_o_id_4;

int             c_id;
       /* 97-02-18          short -> int   */
int             c_id_1, c_id_2, c_id_3, c_id_4;       /*
97-02-18 3  4                              */
short  c_d_id;
int             c_d_id_1, c_d_id_2;
short  c_w_id;
int             c_w_id_1, c_w_id_2;
char   c_first[17];
char   c_middle[3];
char   c_last[17];
char   c_street_1[21];
char   c_street_2[21];
char   c_city[21];
char   c_state[3];
char   c_zip[10];
char   c_phone[17];       /* 1997.02.27 */
char   c_since[15];
char   c_credit[3];
float  c_credit_lim;
int             c_credit_lim_1, c_credit_lim_2,
c_credit_lim_3, c_credit_lim_4;
```

```
int             c_credit_lim_5, c_credit_lim_6,
c_credit_lim_7;
int     c_discount;
int             c_discount_1, c_discount_2;
float   c_balance;
int             c_balance_1, c_balance_2,
c_balance_3, c_balance_4;
int             c_balance_5, c_balance_6,
c_balance_7;
float   c_ytd_payment;
int             c_ytd_payment_1, c_ytd_payment_2,
c_ytd_payment_3, c_ytd_payment_4;
int             c_ytd_payment_5, c_ytd_payment_6,
c_ytd_payment_7;
int     c_payment_cnt;
int             c_payment_cnt_1, c_payment_cnt_2;
int     c_delivery_cnt;
int             c_delivery_cnt_1, c_delivery_cnt_2;
char    c_data[501];

int             h_c_id;
                /* 97-02-18         short -> int   */
int             h_c_id_1, h_c_id_2,h_c_id_3,
h_c_id_4;   /* 97-02-18 3  4                    */
short h_c_d_id;
int             h_c_d_id_1, h_c_d_id_2;
short h_c_w_id;
int             h_c_w_id_1, h_c_w_id_2;
short h_d_id;
int             h_d_id_1, h_d_id_2;
short h_w_id;
int             h_w_id_1, h_w_id_2;
char    h_date[15];
int     h_amount;
int             h_amount_1, h_amount_2,
h_amount_3, h_amount_4;
char            h_data[25];

int     o_id;
int             o_id_1, o_id_2, o_id_3, o_id_4;

short o_d_id;
int             o_d_id_1, o_d_id_2;
short o_w_id;
int             o_w_id_1, o_w_id_2;
int     o_c_id;
        /* 97-02-18         short -> int   */
int             o_c_id_1, o_c_id_2, o_c_id_3,
o_c_id_4;   /* 97-02-18 3  4                    */
char    o_entry_d[15];
short o_carrier_id;
int             o_carrier_id_1, o_carrier_id_2;
short o_ol_cnt;
int             o_ol_cnt_1, o_ol_cnt_2;
short o_all_local;
int             o_all_local_1, o_all_local_2;

int     ol_o_id;
int             ol_o_id_1, ol_o_id_2, ol_o_id_3,
ol_o_id_4;
short ol_d_id;
int             ol_d_id_1, ol_d_id_2;
short ol_w_id;
int             ol_w_id_1, ol_w_id_2;
short ol_number;
int             ol_number_1, ol_number_2;
int     ol_i_id;
int             ol_i_id_1, ol_i_id_2, ol_i_id_3,
ol_i_id_4;
short           ol_supply_w_id;
int             ol_supply_w_id_1, ol_supply_w_id_2;
char    ol_delivery_d[15];
int     ol_quantity;
int             ol_quantity_1, ol_quantity_2;
int     ol_amount;
int             ol_amount_1, ol_amount_2,
ol_amount_3, ol_amount_4;
char    ol_dist_info[25];

int             no_o_id;
int             no_o_id_1, no_o_id_2, no_o_id_3,
no_o_id_4;
short no_d_id;
int             no_d_id_1, no_d_id_2;
short no_w_id;
int             no_w_id_1, no_w_id_2;

/*short         c;              */      /* NURand
*/
short ocid[CUST_PER_DIST];      /* o_c_id
*/
short counter;              /* o_c_id               */

/*      :961127:K.Fukui: I_ID           (main      ) */
        char  *EnvGetI_ID;
        int             I_ID_Rand_by;
/*      :961127:K.Fukui: (above is all) */

void    item();
void    warehouse();
void    stock();
void    district();
void    customer();
void    history();      /* 1997.02.25     */
void    orders();
void    make_address();
void    lastname();
int             make_alpha_string();
int             make_number_string();
#ifdef call_rand
int             random_number();
#else
#define random_number(x,y)
((int)(lrand48()%(y-x+1)) + x)
#endif
void    set_seed();
int             nurand();
void    init_permutation();
/*int   get_permutation();*/

/*              */
FILE  *fst1;
FILE  *fst2;
FILE  *fst3;
FILE  *fst4;
FILE  *fst5;
FILE  *fst6;
FILE  *fst7;
FILE  *fst8;
FILE  *fst9;
int     wst;
int     op_item     ;           /* 1997.02.24 open+write
*/
int     op_stock    ;           /* 1997.02.24
open+write */
int     op_customer  ;          /* 1997.02.24
open+write */
int     op_history   ;          /* 1997.02.24
open+write */
int     op_orders    ;          /* 1997.02.24
open+write */
int     op_orderline ;          /* 1997.02.24
open+write */

char    fileout[100];           /*              */
char    filedum[100];

/*
 *      Function     : main()
 *      Description : DB                , item,
warehouse           ,
 *                                      .
 *      Parameters : 1. argc,
 *                              2. argv,
 *
 *
 *      Grobals Ref: nothing
 *      Grobals Out         : 1. yyyymmddhhmmss,
 *      Returns             : 0
 *                      1
 */

int
main(argc, argv)
int     argc;
char    **argv;
{
        time_t                  tod;    /*                      */
        struct tm     *stm; /*                       */
        int     count_ware;
        int     last_ware;
        int     base_ware;
        int make_type;
        int     mk_loop;
        char    sw_buf[1];

        /*                      */
        if (argc < 5) {
                printf("usage: wttpcc [output_dir] "
              "[start_warehouse] "
              "[end_warehouse] "
              "[maketype]..\n\n");
                printf(" [maketype] make data seeds
for rdbsloader "
              "(multiple designation available)\n");
                printf("          I:Item, D:District,
W:Warehouse, S:Stock,\n");
                printf("          C:Customer, H:History,"
                    "
O:Orders/OrderLine/NewOrder\n");
                exit(1);
        }

        /*                      */
        strcpy( fileout, argv[1] );
        base_ware = atoi(argv[2]);
        last_ware = atoi(argv[3]);
    /*  make_type = atoi(argv[3]); */
        count_ware = last_ware - base_ware;
        /*if (count_ware <= 0) {   1997.02.21 */
        if (count_ware < 0) {
                printf("%s: invalid warehouse
count\n",argv[0]);
```

```
            exit(1);
        }


    /*  system("date"); */
    /*  printf("%s start\n",argv[0]); */
        printf("wttpcc:    :            : (%d-%dWH)
",base_ware,last_ware);
        for( mk_loop = 4; mk_loop <= argc-1;
mk_loop++ ){

            strcpy(sw_buf, argv[mk_loop]);

            switch( sw_buf[0] ){
                case 'I':    printf("Item(%3d),
",ITEM_SIZE);
                break;
                case 'W':
        printf("Warehouse(%3d),
",WAREHOUSE_SIZE);
                break;
                case 'S':    printf("Stock(%3d),
",STOCK_SIZE);
                break;
                case 'D':    printf("District(%3d),
",DISTRICT_SIZE);
                break;
                case 'C':
        printf("Customer(%3d),
",CUSTOMER_SIZE);
                break;
                case 'H':    printf("History(%3d),
",HISTORY_SIZE);
                break;
                case 'O':    printf("Orders(%3d),
N.Order(%3d), O.Line(%3d), ",
                    ORDERS_SIZE,NEWORDE
R_SIZE,ORDERLINE_SIZE);
                break;
            }
        }
        printf("to %s\n", fileout);

    /*    :961127:K.Fukui: ITEM_ID        (?)(     1
10    n
                I_ID             )

                    "TPCRANDBY"    integer n
(100000/n  NURand      ,
                    *n    )              .
                ( 1 < TPCRANDBY < MAXITEMS
"1"         )   */


        EnvGetI_ID = getenv( "TPCRANDBY" );

        if( EnvGetI_ID == NULL ){
            I_ID_Rand_by = 1;
            printf("wttpcc:    : "
          "TPCRANDBY:                OL_I_ID
\n");

        } else {

            I_ID_Rand_by = atoi( EnvGetI_ID );
            printf("wttpcc:    : "
          "TPCRANDBY: %d        OL_I_ID
\n", I_ID_Rand_by);

            if( ( I_ID_Rand_by < 1 ) ||
( I_ID_Rand_by > MAXITEMS ) ){
                I_ID_Rand_by = 1;
                printf("wttpcc:    : "
            "TPCRANDBY:          (1  %d)."
            "      OL_I_ID            \n",
            MAXITEMS);

            }

        }

    /*    :961127:K.Fukui: (above is all) */

        /*                */
        set_seed(time(0));

    /* 1997-02-18 TAB ID 221(c_last NURand     C)
*/
        /* NURand                    */
        /* c = random_number(0, 255);   */
        printf("wttpcc:    : TAB ID 221     C-Delta
= %d \n",C_DELTA) ;
    printf("             C-Load  NURAND       C = %d
\n",C_LOAD ) ;
    printf("             C-Run   NURAND       C = %d
\n",C_RUN  ) ;

        /*                  */
        time(&tod);
        stm = localtime(&tod);
        sprintf(yyyymmddhhmmss,"%04d%02d%02
d%02d%02d%02d",
                    stm->tm_year+1900,stm-
>tm_mon+1,stm->tm_mday,
                    stm->tm_hour,stm-
>tm_min,stm->tm_sec);

        for( mk_loop = 4; mk_loop <= argc-1;
mk_loop++ ){

            strcpy(sw_buf, argv[mk_loop]);

            switch( sw_buf[0] ){
                case 'I':
                    if ( base_ware == 1 )
            {

        fprintf(stderr,"wttpcc:    : ITEM
\n");
                                /* item             */
                                item();

        fprintf(stderr,"wttpcc:    : ITEM
\n");
                                }
            else
             {
               printf("wttpcc:    : "
                    "    warehouse  '1'          ,"
                    "ITEM                \n") ;
             }
                    break;

                case 'W':
                    fprintf(stderr,"wttpcc:    : "
            "WAREHOUSE (%d  %dwh)
\n",
                    base_ware,last_ware);
                        /* warehouse            */

        warehouse(base_ware,last_ware);
                        fprintf(stderr,"wttpcc:    : "
                "WAREHOUSE (%d  %dwh)
\n",
                    base_ware,last_ware);
                        break;

                    case 'S':
                        fprintf(stderr,"wttpcc:    : "
                "STOCK (%d  %dwh)              \n",
                    base_ware, last_ware);
                        /* stock            */

        stock(base_ware,last_ware);
                        fprintf(stderr,"wttpcc:    : "
                "STOCK (%d  %dwh)
\n",
                    base_ware, last_ware);
                        break;

                    case 'D':
                        fprintf(stderr,"wttpcc:    : "
                "DISTRICT (%d  %dwh)
\n",
                    base_ware, last_ware);
                        /* district             */

        district(base_ware,last_ware);
                        fprintf(stderr,"wttpcc:    : "
                "DISTRICT (%d  %dwh)
\n",
                    base_ware, last_ware);
                        break;

                    case 'C':
                        fprintf(stderr,"wttpcc:    : "
                "CUSTOMER (%d  %dwh)
\n",
                    base_ware, last_ware);
                        /* customer             */

        customer(base_ware,last_ware);
                        fprintf(stderr,"wttpcc:    : "
                "CUSTOMER (%d  %dwh)
\n",
                    base_ware, last_ware);
                        break;

                    case 'H':
                        fprintf(stderr,"wttpcc:    : "
                "HISTORY (%d  %dwh)
\n",
                    base_ware, last_ware);
                        /* history              */

        history(base_ware,last_ware);
                        fprintf(stderr,"wttpcc:    : "
                "HISTORY (%d  %dwh)
\n",
                    base_ware, last_ware);
                        break;

                    case 'O':
```

```
                        fprintf(stderr,"wttpcc:     : "
                "ORDERS/O.LINE/N.ORDER
(%d  %dwh)              \n",
                base_ware, last_ware);
                        /* orders              */

        orders(base_ware,last_ware);
                        fprintf(stderr,"wttpcc:     : "
                "ORDERS/O.LINE/N.ORDER
(%d  %dwh)              \n",
                base_ware, last_ware);
            }
        }

/*      system("date"); */

        /*              */
        return(0);
}

/*
 *      Function     : item()
 *      Description : item                  .
 *      Parameters : nothing
 *      Grobals Ref: nothing
 *      Grobals Out         : nothing
 *      Returns             : nothing
 */

void
item()
{
        short           idatasiz;
        short           orig[MAXITEMS];
        int                     pos;
        int                     cnt;
        long            d_100 = 100.0;
        /* 1997-02-18  fprintf -> sprintf + fwrite      */
    int         item_lpcnt                      ; /*
*/
    char        *item_ap                        ; /*
*/
    item_str    *item_cp                        ; /*
*/

        /*                  */
        sprintf( filedum, "%s/data", fileout );

        /*if ((fst1 = fopen( filedum ,
"w"))==NULL){   1997.02.24   */
        if ((op_item = open( filedum ,
O_WRONLY|O_CREAT|O_TRUNC,
                S_IRUSR|S_IWUSR|S_IRGRP|S_IW
GRP|S_IROTH ))==NULL){
                printf("wttpcc:       : %s:             \n",
filedum);
                exit(1);
        }

        /* 1997-02-18  fprintf -> sprintf + fwrite      */
    /* ITEM                                 */
    item_ap = (char
*)malloc((size_t)ITEM_SIZE*ITEM_COUNT);
    if ( item_ap == NULL )                      /*
*/
      {                                 /*
*/
        printf("Malloc failed.(item)\n")        ; /*
*/
        exit(1)                         ; /*
*/
      }                                 /*
*/
    item_cp     = (item_str *)item_ap           ; /*
*/
    item_lpcnt = 0                      ; /*
*/

        /*      orig  MAXITEMS          , i_data
"ORIGINAL"
                10                      */
        memset(orig, 0, sizeof(orig));
        for (cnt = 0; cnt < (MAXITEMS / 10); cnt++)
{
                do {
                        pos = random_number(1,
MAXITEMS);
                } while (orig[pos - 1]);
                orig[pos - 1] = 1;
        }

        /* i_id  1-MAXITEMS            , MAXITEMS
*/
        /* item                         */
        for (i_id = 1; i_id <= MAXITEMS; i_id++) {

                /* i_name          */
                make_alpha_string(14, 24, i_name);

                /* i_data       ,     10%       ORIGINAL
*/
                idatasiz = make_alpha_string(26, 50,
i_data);
                if (orig[i_id - 1]) {
                        pos = random_number(0,
idatasiz - 8);
                        strncpy(&i_data[pos],
"ORIGINAL", 8);
                }

                /* record    :     : 1997-02-27 */
        memset(item_cp->i_name, ' '  ,24)     ;
        len_i = strlen(i_name)              ;
        strncpy(item_cp->i_name,i_name,len_i) ;
        memset(item_cp->i_data, ' '  ,50)     ;
        len_i = strlen(i_data)              ;
        strncpy(item_cp->i_data,i_data,len_i) ;

                /* record    :     : 96/09/06      */
        item_cp->i_id_1 = i_id / T256;
                item_cp->i_id_2 = (i_id - ( i_id_1 *
T256 )) / D256;
                item_cp->i_id_3 = (i_id - ( i_id_1 *
T256 )
                        - ( i_id_2 * D256 )) / 256 ;
                item_cp->i_id_4 = i_id % T256 ;

            /* i_im_id      : 97-02-18 start */
                i_im_id = random_number(1, 10000);
                item_cp->i_im_id_1 = i_im_id / T256;
                item_cp->i_im_id_2 = (i_im_id -
(i_im_id_1 * T256)) / D256;
                item_cp->i_im_id_3 = (i_im_id -
(i_im_id_1 * T256)
                        - (i_im_id_2 * D256)) /
256;
                item_cp->i_im_id_4 = i_im_id % T256;
                /* i_im_id      : 97-02-18 end */

                /* i_price          */
        /* i_price /= d_100; */
                i_price = random_number(100,
10000);
                item_cp->i_price_1 = i_price / 256 ;
                item_cp->i_price_2 = i_price % 256 ;

        item_cp     = item_cp     + 1 ;
        item_lpcnt = item_lpcnt + 1 ;

        if ( item_lpcnt == ITEM_COUNT )
         {
         write(op_item,
                item_ap,
                (size_t)ITEM_SIZE *
(size_t)ITEM_COUNT ) ;
           item_cp     = (item_str *)item_ap      ;
           item_lpcnt = 0           ;
         }
        }

        /* 1997-02-18  fprintf -> sprintf + fwrite   */
        if ( item_lpcnt != 0 )
         {
         write(op_item,
                item_ap,
                (size_t)ITEM_SIZE *
(size_t)item_lpcnt ) ;
         }

        /*                  */
        close(op_item);

    /*                  */
    free(item_ap) ;

        /*                  */
        return;
}

/*
 *      Function     : warehouse()
 *      Description : warehouse                     .
 *      Parameters : 1. base_ware,
 *      Parameters : 2. last_ware,
 *      Grobals Ref: nothing
 *      Returns             : nothing
 */

void
warehouse(base_ware,last_ware)
int     base_ware;
int     last_ware;
{

        /*      */
        int             filecount = 1;
        int             outfilecount;
        char            filename[64];

        long            d_10000 = 10000.0;
/*      w_ytd = 300000.00;      record
*/
```

```
        w_ytd = 30000000;       /* 98-11-06 */
        outfilecount = ((base_ware-1)/10) + 1;

        /*                  */
        sprintf(filename , "%s/WH%d_%d", fileout,
base_ware, last_ware);
        if ((fst2 = fopen(filename , "w"))==NULL){
                printf("wttpcc:       : %s:
\n",filename);
                exit(1);
        }

        /* w_id                     , count_ware      */
        /* warehouse                                  */
        for (w_id = base_ware; w_id <= last_ware;
w_id++) {

                /* w_name          */
                make_alpha_string(6, 10, w_name);

                /*                      */
                make_address(w_street_1,
w_street_2, w_city, w_state, w_zip);

        /* w_tax /= d_10000;*/
                w_tax = random_number(0, 2000);

                /* record    :     : 96/09/06 fukui */

                w_id_1 = w_id / 256;
                w_id_2 = w_id % 256;
                w_tax_1 = w_tax / 256;
                w_tax_2 = w_tax % 256;
                w_ytd_1 = 0x00;  /* w_ytd:
+300000.00 */
                w_ytd_2 = 0x00;
                w_ytd_3 = 0x03;
                w_ytd_4 = 0x00;
                w_ytd_5 = 0x00;
                w_ytd_6 = 0x00;
                w_ytd_7 = 0x0c;

                fprintf(fst2 ,
                  "%c%c"
                  "%-10s"
                  "%-20s"
                  "%-20s"
                  "%-20s"
                  "%-2s"
                  "%-9s"
                  "%c%c"
                  "%c%c%c%c%c%c%c%c",
                   w_id_1,w_id_2,
                   w_name,
                   w_street_1,w_street_2,
                   w_city,
                   w_state,
                   w_zip,
                   w_tax_1,w_tax_2,
                   w_ytd_1,w_ytd_2,w_ytd_3,w_ytd_4,
w_ytd_5,w_ytd_6,w_ytd_7);

                filecount++;

        }

        /*                  */
        fclose(fst2);

        /*                  */
        return;

}

/*
 *      Function     : stock()
 *      Description : stock                    .
 *      Parameters : 1. base_ware,
 *      Parameters : 2. last_ware,
 *      Grobals Ref: nothing
 *      Grobals Out         : nothing
 *      Returns             : nothing
 */

void
stock(base_ware,last_ware)
int     base_ware;
int     last_ware;
{
        /*       */
        short  sdatasiz;
        short  orig[MAXITEMS];
        int             pos;
        int             cnt;
        int             filecount = 1;
        int             outfilecount;
        char   filename[64];
        /* 1997-02-18  fprintf -> sprintf + fwrite      */
    int         stock_lpcnt                     ; /*
*/
    char       *stock_ap                        ; /*
*/
    stock_str  *stock_cp                        ; /*
*/

        s_ytd = 0;
        s_order_cnt = 0;
        s_remote_cnt = 0;
        outfilecount = ((base_ware-1)/10) + 1;

        /*                  */
        sprintf(filename , "%s/ST%d_%d" ,fileout,
base_ware, last_ware);
        /*if ((fst3 = fopen(filename ,
"w"))==NULL){       1997.02.24 */
        if ((op_stock = open( filename ,
O_WRONLY|O_CREAT|O_TRUNC,
                S_IRUSR|S_IWUSR|S_IRGRP|S_IW
GRP|S_IROTH ))==NULL){
                printf("wttpcc:       : %s:              \n" ,
filename);
                exit(1);
        }

        /* 1997-02-18  fprintf -> sprintf + fwrite      */
    /* STOCK                                    */
    stock_ap = (char
*)malloc((size_t)STOCK_SIZE*STOCK_COUNT);
    if ( stock_ap == NULL )                     /*
*/
      {                                         /*
*/
        printf("Malloc failed.(stock)\n")       ; /*
*/
        exit(1)                                 ; /*
*/
    }                                           /*
*/
    stock_cp    = (stock_str *)stock_ap         ; /*
*/
    stock_lpcnt = 0                             ; /*
*/

        /* w_id                 count_ware              */
        for (s_w_id = base_ware; s_w_id <=
last_ware; s_w_id++){

                fprintf(stderr,"wttpcc:     : "
                   "STOCK %d/%d  %d      \n" ,
                   s_w_id, base_ware, last_ware);

                /*     orig  MAXSTOCK
           10                                   */
                memset(orig, 0, sizeof(orig));
                for (cnt = 0; cnt < (MAXSTOCK / 10);
cnt++) {
                        do {
                                pos = random_number(1,
MAXSTOCK);
                        } while (orig[pos - 1]);
                        orig[pos - 1] = 1;
                }

                /* s_i_id               , MAXSTOCK     */
                /* stock                                */
                for (s_i_id = 1; s_i_id <= MAXSTOCK;
s_i_id++) {

                        /* s_quantity          */
                        s_quantity =
random_number(10, 100);

                        /* s_dist_01 .. s_dist_10
*/
                        make_alpha_string(24, 24,
s_dist_01);
                        make_alpha_string(24, 24,
s_dist_02);
                        make_alpha_string(24, 24,
s_dist_03);
                        make_alpha_string(24, 24,
s_dist_04);
                        make_alpha_string(24, 24,
s_dist_05);
                        make_alpha_string(24, 24,
s_dist_06);
                        make_alpha_string(24, 24,
s_dist_07);
                        make_alpha_string(24, 24,
s_dist_08);
                        make_alpha_string(24, 24,
s_dist_09);
                        make_alpha_string(24, 24,
s_dist_10);

                        /* s_data       ,     10%
ORIGINAL         */
                        sdatasiz =
make_alpha_string(26, 50, s_data);
                        if (orig[s_i_id - 1]) {
                                pos = random_number(0,
sdatasiz - 8);
                                strncpy(&s_data[pos],
"ORIGINAL", 8);
```

```
                }

                /* record    :    : 1997-02-27 */
        strncpy(stock_cp-
>s_dist_01,s_dist_01,24) ;
        strncpy(stock_cp-
>s_dist_02,s_dist_02,24) ;
        strncpy(stock_cp-
>s_dist_03,s_dist_03,24) ;
        strncpy(stock_cp-
>s_dist_04,s_dist_04,24) ;
        strncpy(stock_cp-
>s_dist_05,s_dist_05,24) ;
        strncpy(stock_cp-
>s_dist_06,s_dist_06,24) ;
        strncpy(stock_cp-
>s_dist_07,s_dist_07,24) ;
        strncpy(stock_cp-
>s_dist_08,s_dist_08,24) ;
        strncpy(stock_cp-
>s_dist_09,s_dist_09,24) ;
        strncpy(stock_cp-
>s_dist_10,s_dist_10,24) ;

        memset(stock_cp->s_data,' ',50)          ;
        len_i = strlen(s_data)                   ;
        strncpy(stock_cp->s_data,s_data,len_i)   ;

                /* record    :    : 96/09/09 fukui
*/
                stock_cp->s_i_id_1 = s_i_id /
T256;
                stock_cp->s_i_id_2 = (s_i_id -
(s_i_id_1 * T256)) / D256;
                stock_cp->s_i_id_3 = (s_i_id -
(s_i_id_1 * T256)
                        - (s_i_id_2 * D256)) /
256;
                stock_cp->s_i_id_4 = s_i_id %
T256;
                stock_cp->s_w_id_1 = s_w_id /
256;
                stock_cp->s_w_id_2 =
s_w_id % 256;
                stock_cp->s_quantity_1 =
s_quantity / 256;
                stock_cp->s_quantity_2 =
s_quantity % 256;
                stock_cp->s_ytd_1 = s_ytd /
T256;
                stock_cp->s_ytd_2 = (s_ytd -
(s_ytd_1*T256)) / D256;
                stock_cp->s_ytd_3 = (s_ytd -
(s_ytd_1*T256)-(s_ytd_2*D256)) / 256;
                stock_cp->s_ytd_4 = s_ytd %
T256;
                stock_cp->s_order_cnt_1 =
s_order_cnt / 256;
                stock_cp->s_order_cnt_2 =
s_order_cnt % 256;
                stock_cp->s_remote_cnt_1 =
s_remote_cnt / 256;
                stock_cp->s_remote_cnt_2 =
s_remote_cnt % 256;

        stock_cp    = stock_cp    + 1 ;
        stock_lpcnt = stock_lpcnt + 1 ;

        if ( stock_lpcnt == STOCK_COUNT )
         {
         write(op_stock,
               stock_ap,
               (size_t)STOCK_SIZE *
(size_t)STOCK_COUNT ) ;
            stock_cp    = (stock_str *)stock_ap    ;
            stock_lpcnt = 0           ;
          }
             }

            filecount++;
        }

        /* 1997-02-18  fprintf -> sprintf + fwrite   */
       if ( stock_lpcnt != 0 )
        {
        write(op_stock,
              stock_ap,
              (size_t)STOCK_SIZE *
(size_t)stock_lpcnt ) ;
        }

        /*                */
        close(op_stock);

    /*                */
    free(stock_ap) ;

        /*                */
        return;

}

/*
 *      Function     : district()
 *      Description : district                .
 *      Parameters : 1. base_ware,
 *      Parameters : 2. last_ware,
 *      Grobals Ref: nothing
 *      Grobals Out        : nothing
 *      Returns            : nothing
 */

void
district(base_ware,last_ware)
int     base_ware;
int     last_ware;
{
        /*      */
        long   d_10000 = 10000.0;
        int            filecount = 1;
        int            outfilecount;
        char   filename[64];

/*      d_ytd = 30000.00; record                */
        d_ytd = 3000000; /* 98-11-06 */
        d_next_o_id = 3001;
        outfilecount = ((base_ware-1)/10) +1;

        /*                */
        sprintf(filename , "%s/DI%d_%d" ,fileout,
base_ware, last_ware);
        if ((fst4 = fopen(filename , "w"))==NULL){
                printf("wttpcc:       : %s:             \n" ,
filename);
                exit(1);
        }

        /* w_id             , count_ware          */
        for (d_w_id = base_ware; d_w_id <=
last_ware; d_w_id++){

            /* d_id             , DIST_PER_WARE
*/
                for (d_id = 1; d_id <=
DIST_PER_WARE; d_id++) {

                        /* d_name          */
                        make_alpha_string(6, 10,
d_name);

                        /*                      */
                        make_address(d_street_1,
d_street_2, d_city, d_state, d_zip);

                        /* d_tax          */
                /* d_tax /= d_10000;*/
                        d_tax = random_number(0,
2000);

                        /* record    :    : 96-09-06 fukui
*/
                        d_id_1 = d_id / 256;
                        d_id_2 = d_id % 256;
                        d_w_id_1 = d_w_id / 256;
                        d_w_id_2 = d_w_id % 256;
                        d_tax_1 = d_tax / 256;
                        d_tax_2 = d_tax % 256;
                        d_ytd_1 = 0x00;   /* d_ytd:
30000.00 */
                        d_ytd_2 = 0x00;
                        d_ytd_3 = 0x00;
                        d_ytd_4 = 0x30;
                        d_ytd_5 = 0x00;
                        d_ytd_6 = 0x00;
                        d_ytd_7 = 0x0c;
                        d_next_o_id_1 = d_next_o_id /
T256;
                        d_next_o_id_2 = (d_next_o_id-
(d_next_o_id_1*T256))/D256;
                        d_next_o_id_3 = (d_next_o_id-
(d_next_o_id_1*T256)
                                        -
(d_next_o_id_2*D256))/256;
                        d_next_o_id_4 = d_next_o_id %
T256;

                        fprintf(fst4 ,"%c%c"
                        "%c%c"
                        "%-10s"
                        "%-20s%-20s%-20s%-2s%-9s"
                        "%c%c"
                        "%c%c%c%c%c%c%c%c"
                        "%c%c%c%c",
                    d_id_1,d_id_2,
                    d_w_id_1,d_w_id_2,
                    d_name,
                    d_street_1,d_street_2,d_city,d_stat
e,d_zip,
                    d_tax_1,d_tax_2,
                    d_ytd_1,d_ytd_2,d_ytd_3,d_ytd_4,d
_ytd_5,d_ytd_6,d_ytd_7,
                    d_next_o_id_1,d_next_o_id_2,d_ne
xt_o_id_3,d_next_o_id_4);
```

```
                }
                filecount++;
        }

        /*              */
        fclose(fst4);

        /*              */
        return;

}

/*
 *      Function     : customer()
 *      Description : customer                  .
 *      Parameters : 1. base_ware,
 *      Parameters : 2. last_ware,
 *      Grobals Ref: yyyymmddhhmmss,
 *      Grobals Out         : nothing
 *      Returns             : nothing
 */

void
customer(base_ware,last_ware)
int     base_ware;
int     last_ware;
{
        /*      */
        long            d_10000 = 1000.0;
        long            d_100 = 100.0;
        int                     filecount = 1;
        int                     outfilecount;
        char            filename1[64];
        /* 1997-02-18  fprintf -> sprintf + fwrite      */
    int         customer_lpcnt              ; /*
*/
    char        *customer_ap                ; /*
*/
    customer_str *customer_cp               ; /*
*/

/*      c_credit_lim = 50000.00;        record       ,
*/
        c_credit_lim = 5000000;         /* 1998.11.06
*/
/*      c_balance = -10.00;
record       ,          */
        c_balance = -1000;                      /*
1998.11.06  */
/*      c_ytd_payment = 10.00;          record       ,
*/
        c_ytd_payment = 1000;           /* 1998.11.06
*/
        c_payment_cnt = 0;          /* 1997.03.05
*/
        c_delivery_cnt = 0;
        strcpy(c_middle, "OE");
        strcpy(c_since, yyyymmddhhmmss);

        outfilecount = ((base_ware-1)/10) +1;
        /*              */
        sprintf(filename1 , "%s/CU%d_%d" ,fileout,
base_ware, last_ware);
        /*if ((fst5 = fopen(filename1 ,
"w"))==NULL){   1997.02.24 */
        if ((op_customer =
open( filename1,O_WRONLY|O_CREAT|O_TRUN
C,
                S_IRUSR|S_IWUSR|S_IRGRP|S_IW
GRP|S_IROTH ))==NULL){
                printf("wttpcc:       : %s:             \n" ,
filename1);
                exit(1);
        }

        /* 1997-02-18  fprintf -> sprintf + fwrite      */
    /* CUSTOMER                                 */
    customer_ap = (char
*)malloc((size_t)(CUSTOMER_SIZE*CUSTOMER
_COUNT) ) ;
    if ( customer_ap == NULL )                  /*
*/
      {                                         /*
*/
        printf("Malloc failed.(customer)\n")    ; /*
*/
        exit(1)                                 ; /*
*/
      }                                         /*
*/
    customer_cp    = (customer_str
*)customer_ap  ; /*                      */
    customer_lpcnt = 0                          ; /*
*/

        /* w_id                 count_ware          */
        for (c_w_id = base_ware; c_w_id <=
last_ware; c_w_id++){

                fprintf(stderr,"wttpcc:     :
CUSTOMER %d/%d  %d       \n"
                                                ,c_w_id,
base_ware,last_ware);

                /* d_id                 DIST_PER_WARE
*/
                for (c_d_id = 1; c_d_id <=
DIST_PER_WARE; c_d_id++){

                        /* c_id
CUST_PER_DIST           */
                        /* coustomer, history
*/
                        for (c_id = 1; c_id <=
CUST_PER_DIST; c_id++) {

                                make_alpha_string(8, 16,
c_first)    ;

                /* 1997-02-18 TAB ID 221(c_last
NURand     C)           */
                                /*      1000            ,
2000       */
                                /*          lastname
c_last          */
                                if (c_id <= 1000) {
                                        lastname(c_id - 1,
c_last);
                                } else {

        lastname(nurand(255, 0, 999,C_LOAD),
c_last);
                                }
                                /*      ,                       */
                                make_address(c_street_1,
c_street_2, c_city, c_state, c_zip);

                                make_number_string(16,
16, c_phone);

                                /* c_credit          10%
BC, 90%  GC          */
                                if (random_number(0, 9))
{
                                        strcpy(c_credit,
"GC");
                                } else {
                                        strcpy(c_credit,
"BC");
                                }

                                /* c_discount          */
                        /* c_discount /=  d_10000;*/
                                c_discount =
random_number(0, 5000);

                                /* c_data          */
                                make_alpha_string(300,
500, c_data);

                                /* record    :    : 1997-02-
27 */
                        memset(customer_cp , ' ' ,
CUSTOMER_SIZE)        ;
                        len_i = strlen(c_first)                         ;
                        strncpy(customer_cp-
>c_first   ,c_first   ,len_i) ;
                        strncpy(customer_cp-
>c_middle  ,c_middle  ,2    ) ;
                        len_i = strlen(c_last)                          ;
                        strncpy(customer_cp-
>c_last    ,c_last    ,len_i) ;

                        len_i =
strlen(c_street_1)                      ;
                        strncpy(customer_cp-
>c_street_1,c_street_1,len_i) ;
                        len_i =
strlen(c_street_2)                      ;
                        strncpy(customer_cp-
>c_street_2,c_street_2,len_i) ;
                        len_i = strlen(c_city)                          ;
                        strncpy(customer_cp-
>c_city    ,c_city    ,len_i) ;
                        strncpy(customer_cp-
>c_state   ,c_state   ,2    ) ;
                        strncpy(customer_cp-
>c_zip     ,c_zip     ,9    ) ;
                        strncpy(customer_cp-
>c_phone   ,c_phone   ,16   ) ;
                        strncpy(customer_cp-
>c_since   ,c_since   ,14   ) ;
                        strncpy(customer_cp-
>c_credit  ,c_credit  ,2    ) ;

                        len_i =
strlen(c_data)                          ;
                        strncpy(customer_cp-
>c_data    ,c_data    ,len_i) ;
```

```
                        /* record    :     : 96/09/09
fukui */
        /*
                        c_id_1 = c_id / 256;
                        c_id_2 = c_id % 256;
        */
        /* 97-02-18 c_id          short -> int */
                        customer_cp->c_id_1 =
c_id / T256;
                        customer_cp->c_id_2 =
(c_id - (c_id_1 * T256)) / D256;
                        customer_cp->c_id_3 =
(c_id - (c_id_1 * T256)
                              - (c_id_2 * D256)) /
256;
                        customer_cp->c_id_4 =
c_id % T256;

                        customer_cp->c_d_id_1 =
c_d_id / 256;
                        customer_cp->c_d_id_2 =
c_d_id % 256;
                        customer_cp->c_w_id_1
= c_w_id / 256;
                        customer_cp->c_w_id_2
= c_w_id % 256;
                        customer_cp-
>c_credit_lim_1 = 0x00;/*
c_credit_lim:+50000.00*/
                        customer_cp-
>c_credit_lim_2 = 0x00;
                        customer_cp-
>c_credit_lim_3 = 0x00;
                        customer_cp-
>c_credit_lim_4 = 0x50;
                        customer_cp-
>c_credit_lim_5 = 0x00;
                        customer_cp-
>c_credit_lim_6 = 0x00;
                        customer_cp-
>c_credit_lim_7 = 0x0c;
                        customer_cp-
>c_discount_1 = c_discount / 256;
                        customer_cp-
>c_discount_2 = c_discount % 256;
                        customer_cp-
>c_balance_1 = 0x00;          /* c_balance: -10.00
*/
                        customer_cp-
>c_balance_2 = 0x00;
                        customer_cp-
>c_balance_3 = 0x00;
                        customer_cp-
>c_balance_4 = 0x00;
                        customer_cp-
>c_balance_5 = 0x01;
                        customer_cp-
>c_balance_6 = 0x00;
                        customer_cp-
>c_balance_7 = 0x0d;
                        customer_cp-
>c_ytd_payment_1 = 0x00; /*
c_ytd_payment:+10.00*/
                        customer_cp-
>c_ytd_payment_2 = 0x00;
                        customer_cp-
>c_ytd_payment_3 = 0x00;
                        customer_cp-
>c_ytd_payment_4 = 0x00;
                        customer_cp-
>c_ytd_payment_5 = 0x01;
                        customer_cp-
>c_ytd_payment_6 = 0x00;
                        customer_cp-
>c_ytd_payment_7 = 0x0c;
                        customer_cp-
>c_payment_cnt_1 = c_payment_cnt / 256;
                        customer_cp-
>c_payment_cnt_2 = c_payment_cnt % 256;
                        customer_cp-
>c_delivery_cnt_1 = c_delivery_cnt / 256;
                        customer_cp-
>c_delivery_cnt_2 = c_delivery_cnt % 256;

              customer_cp    = customer_cp    + 1 ;
              customer_lpcnt = customer_lpcnt + 1 ;

              if ( customer_lpcnt ==
CUSTOMER_COUNT )
               {
               write(op_customer ,
                     customer_ap ,
                     (size_t)CUSTOMER_SIZE *
(size_t)CUSTOMER_COUNT) ;
                customer_cp    = (customer_str
*)customer_ap ;
                customer_lpcnt = 0          ;
               }

                     }
               }
               filecount++;
        }

        /* 1997-02-18  fprintf -> sprintf + fwrite   */
       if ( customer_lpcnt != 0 )
        {
        write(op_customer ,
              customer_ap ,
              (size_t)CUSTOMER_SIZE *
(size_t)customer_lpcnt) ;
        }

        /*                */
        close(op_customer) ;

    /*                */
    free(customer_ap) ;

        /*                */
        return;

}
/*
 *      Function     : history()
 *      Description : history                .
 *      Parameters : 1. base_ware,
 *      Parameters : 2. last_ware,
 *      Grobals Ref: yyyymmddhhmmss,
 *      Grobals Out        : nothing
 *      Returns            : nothing
 */

void
history(base_ware,last_ware)
int     base_ware;
int     last_ware;
{
        /*       */
        int                filecount = 1;
        int                outfilecount;
        char           filename2[64];
        /* 1997-02-18  fprintf -> sprintf + fwrite      */
    int         history_lpcnt                ; /*
*/
    char        *history_ap                  ; /*
*/
    history_str *history_cp                  ; /*
*/

        h_amount = 10;
        strcpy(h_date, yyyymmddhhmmss);

        outfilecount = ((base_ware-1)/10) +1;
        sprintf(filename2 , "%s/HI%d_%d" ,fileout,
base_ware, last_ware);
        /*if ((fst6 = fopen(filename2 ,
"w"))==NULL){   1997.02.24 */
        if ((op_history =
open( filename2,O_WRONLY|O_CREAT|O_TRUN
C,
                S_IRUSR|S_IWUSR|S_IRGRP|S_IW
GRP|S_IROTH ))==NULL){
                printf("wttpcc:       : %s:              \n" ,
filename2);
                exit(1);
        }

        /* 1997-02-18  fprintf -> sprintf + fwrite      */
    /* HISTORY                                  */
    history_ap = (char
*)malloc((size_t)(HISTORY_SIZE*HISTORY_COU
NT)  );
    if ( history_ap == NULL )                    /*
*/
     {                                   /*
*/
        printf("Malloc failed.(history)\n")      ; /*
*/
        exit(1)                          ; /*
*/
     }                                   /*
*/
    history_cp     = (history_str *)history_ap    ; /*
*/
    history_lpcnt  = 0                           ; /*
*/

        /* w_id                count_ware          */
        for (h_c_w_id = base_ware; h_c_w_id <=
last_ware; h_c_w_id++){

                fprintf(stderr,"wttpcc:     :
HISTORY %d/%d  %d       \n"
                                          ,h_c_w_id,
base_ware,last_ware);

                /* d_id               DIST_PER_WARE
*/
                for (h_c_d_id = 1; h_c_d_id <=
DIST_PER_WARE; h_c_d_id++){
```

```
                        /* c_id
CUST_PER_DIST           */
                        /* coustomer, history
*/
                        for (h_c_id = 1; h_c_id <=
CUST_PER_DIST; h_c_id++) {

                                /*h_c_w_id = c_w_id;*/
                                /*h_c_d_id = c_d_id;*/
                                /*h_c_id = c_id;    */
                                h_w_id = h_c_w_id;
                                h_d_id = h_c_d_id;

                        /* h_data          */
                                make_alpha_string(12, 24,
h_data);

                                /* record    :    : 1997-02-
27 */
                        memset(history_cp->h_data , ' ' ,
24)    ;
                len_i = strlen(h_data)                  ;
                strncpy(history_cp-
>h_data,h_data,len_i) ;
                strncpy(history_cp-
>h_date,h_date,14    ) ;

                /* 97-02-18 h_c_id         short -> int */
                                history_cp->h_c_id_1=
h_c_id / T256;
                                history_cp-
>h_c_id_2=(h_c_id - (h_c_id_1 * T256)) / D256;
                                history_cp-
>h_c_id_3=(h_c_id - (h_c_id_1 * T256)
                                        - (h_c_id_2 * D256))
/256;
                                history_cp->h_c_id_4=
h_c_id % T256;

                                history_cp->h_c_d_id_1 =
h_c_d_id / 256;
                                history_cp->h_c_d_id_2 =
h_c_d_id % 256;
                                history_cp->h_c_w_id_1 =
h_c_w_id / 256;
                                history_cp->h_c_w_id_2 =
h_c_w_id % 256;
                                history_cp->h_d_id_1 =
h_d_id / 256;
                                history_cp->h_d_id_2 =
h_d_id % 256;
                                history_cp->h_w_id_1 =
h_w_id / 256;
                                history_cp->h_w_id_2 =
h_w_id % 256;
                                history_cp->h_amount_1
= h_amount / T256;
                                history_cp->h_amount_2
= (h_amount - (h_amount_1*T256)) / D256;
                                history_cp->h_amount_3
= (h_amount - (h_amount_1*T256)
                                                -
(h_amount_2*D256)) / D256;
                                history_cp->h_amount_4
= h_amount % T256;

                history_cp    = history_cp    + 1 ;
                history_lpcnt = history_lpcnt + 1 ;

                if ( history_lpcnt ==
HISTORY_COUNT )
                        {
                        write(op_history ,
                                history_ap ,
                                (size_t)HISTORY_SIZE  *
(size_t)HISTORY_COUNT) ;
                          history_cp    = (history_str
*)history_ap   ;
                          history_lpcnt = 0           ;
                        }

                                }
                        }
                        filecount++;
                }

                /* 1997-02-18  fprintf -> sprintf + fwrite   */
                if ( history_lpcnt != 0 )
                {
                write(op_history ,
                        history_ap ,
                        (size_t)HISTORY_SIZE  *
(size_t)history_lpcnt) ;
                }

                /*                */
                close(op_history ) ;

        /*                */
        free(history_ap) ;

                /*                */
                return;

}

/*
 *      Function     : orders()
 *      Description : orders, order_line,
new_order                .
 *      Parameters : 1. base_ware,
 *      Parameters : 1. last_ware,
 *      Grobals Ref: yyyymmddhhmmss,
 *      Grobals Out         : nothing
 *      Returns             : nothing
 */

void
orders(base_ware,last_ware)
int     base_ware;
int     last_ware;
{
        /*      */
        double              d_100 = 100;
        int                 filecount = 1;
        int                 outfilecount;
        char            filename1[64];
        char            filename2[64];
        char            filename3[64];
        short           d_id;
        short           w_id;
        int                 o_id;
        /* 1997-02-18  fprintf -> sprintf + fwrite      */
    int         orders_lpcnt                ; /*
*/
    char        *orders_ap                  ; /*
*/
    orders_str  *orders_cp                  ; /*
*/
    int         orderline_lpcnt             ; /*
*/
    char        *orderline_ap               ; /*
*/
    orderline_str *orderline_cp             ; /*
*/

        o_all_local = 1;
        ol_quantity = 5;
        outfilecount = ((base_ware-1)/10) + 1;

        /*                */
        sprintf(filename1 , "%s/OS%d_%d" ,fileout,
base_ware, last_ware);
        /*if ((fst7 = fopen(filename1 ,
"w"))==NULL){   1997.02.24 */
        if ((op_orders =
open( filename1,O_WRONLY|O_CREAT|O_TRUN
C,
                S_IRUSR|S_IWUSR|S_IRGRP|S_IW
GRP|S_IROTH ))==NULL){
                printf("wttpcc:      : %s:              \n" ,
filename1);
                exit(1);
        }
        sprintf(filename2 , "%s/NO%d_%d" ,fileout,
base_ware, last_ware);
        if ((fst8 = fopen(filename2 , "w"))==NULL){
                printf("wttpcc:      : %s:              \n" ,
filename2);
                exit(1);
        }
        sprintf(filename3 , "%s/OL%d_%d" ,fileout,
base_ware, last_ware);
        /*if ((fst9 = fopen(filename3 ,
"w"))==NULL){   1997.02.24 */
        if ((op_orderline =
open( filename3,O_WRONLY|O_CREAT|O_TRUN
C,
                S_IRUSR|S_IWUSR|S_IRGRP|S_IW
GRP|S_IROTH ))==NULL){
                printf("wttpcc:      : %s:              \n" ,
filename3);
                exit(1);
        }

        /* 1997-02-18  fprintf -> sprintf + fwrite      */
    /* ORDERS                                   */
    /* ORDERLINE                                */
    orders_ap = (char
*)malloc((size_t)(ORDERS_SIZE*ORDERS_COU
NT)
                        +
(ORDERLINE_SIZE*ORDERLINE_COUNT));
    if ( orders_ap == NULL )                    /*
*/
      {                                         /*
*/
        printf("Malloc failed.(orders)\n")      ;/*
*/
        exit(1)                                 ;/*
*/
      }                                         /*
*/
```

```
   orderline_ap    = orders_ap +
(ORDERS_SIZE*ORDERS_COUNT) ;
   orders_cp       = (orders_str *)orders_ap      ;/*
*/
   orderline_cp    = (orderline_str *)orderline_ap;/*
*/
   orders_lpcnt    = 0                            ;/*
*/
   orderline_lpcnt = 0                            ;/*
*/

       /* w_id              , count_ware          */
       for (w_id = base_ware; w_id <= last_ware;
w_id++){

                fprintf(stderr,"wttpcc:     :
ORDERS/O.LINE/N.ORDER %d/%d  %d       \n"

                              ,w_id, base_ware,
last_ware);

               /* d_id               , DIST_PER_WARE
*/
               for (d_id = 1; d_id <=
DIST_PER_WARE; d_id++) {

                        /* o_c_id
*/
                        init_permutation();

               /* o_id  1-ORD_PER_DIST               ,
ORD_PER_DIST     */
               /* orders                                */
                        for (o_id = 1; o_id <=
ORD_PER_DIST; o_id++) {

                                no_o_id = o_id;
                                no_w_id = w_id;
                                no_d_id = d_id;
                                o_id = o_id;
                                o_w_id = w_id;
                                o_d_id = d_id;
                                o_ol_cnt =
random_number(5, 15);
                                strcpy(o_entry_d,
yyyymmddhhmmss);
                                ol_o_id = o_id;
                                ol_w_id = w_id;
                                ol_d_id = d_id;
                                ol_supply_w_id = w_id;

                                /* o_c_id
1997-02-18 */
               /* o_c_id          init_permutation
*/
               /*       1
CUST_PER_DIST                          . */
                        /*
*/
                    counter++;
                    /* ocid  counter-1
*/
                                o_c_id = ocid[counter -
1] ;

                                orders_cp->o_entry_d_v1
= NNUL_V;
                                orders_cp->o_entry_d_v2
= NNUL_V;
                    strncpy(orders_cp-
>o_entry_d,o_entry_d,14) ;

                                /* o_id>2100      ,          */
                                /*              ,
new_order */
                                if (o_id >
(ORD_PER_DIST - NEWORDS)){
                                        /* ocarrier_id
*/
                                  /*  o_carrier_id = 0x00;
*/      /* NULL   */
                                        orders_cp-
>o_carrier_id_v1 = NUL_V;
                                        orders_cp-
>o_carrier_id_v2 = NUL_V;
                                        orders_cp-
>o_carrier_id_1 = 0x00;
                                        orders_cp-
>o_carrier_id_2 = 0x00;

                                        /* orders
record    :     : 96/09/09 fukui */
                                        /* (  NULL    ) */
                                        orders_cp->o_id_v1
= NNUL_V;
                                        orders_cp->o_id_v2
= NNUL_V;
                                        orders_cp->o_id_1
= o_id / T256;
                                        orders_cp->o_id_2
= (o_id-(o_id_1*T256))/D256;
                                        orders_cp->o_id_3
= (o_id-(o_id_1*T256)-(o_id_2*D256))/256;
                                        orders_cp->o_id_4
= o_id % T256;

                                        orders_cp-
>o_d_id_v1 = NNUL_V;
                                        orders_cp-
>o_d_id_v2 = NNUL_V;
                                        orders_cp-
>o_d_id_1 = o_d_id / 256;
                                        orders_cp-
>o_d_id_2 = o_d_id % 256;

                                        orders_cp-
>o_w_id_v1 = NNUL_V;
                                        orders_cp-
>o_w_id_v2 = NNUL_V;
                                        orders_cp-
>o_w_id_1 = o_w_id / 256;
                                        orders_cp-
>o_w_id_2 = o_w_id % 256;

                    /* 97-02-18 o_c_id          short->int  */
                                        orders_cp-
>o_c_id_v1 = NNUL_V;
                                        orders_cp-
>o_c_id_v2 = NNUL_V;
                                        orders_cp-
>o_c_id_1 = o_c_id / T256;
                                        orders_cp-
>o_c_id_2 = (o_c_id-(o_c_id_1*T256)) / D256;
                                        orders_cp-
>o_c_id_3 = (o_c_id-(o_c_id_1*T256)
                                                -
(o_c_id_2*D256))/256;
                                        orders_cp-
>o_c_id_4 = o_c_id % T256;

                                        orders_cp-
>o_ol_cnt_v1 = NNUL_V;
                                        orders_cp-
>o_ol_cnt_v2 = NNUL_V;
                                        orders_cp-
>o_ol_cnt_1 = o_ol_cnt / 256;
                                        orders_cp-
>o_ol_cnt_2 = o_ol_cnt % 256;

                                        orders_cp-
>o_all_local_v1 = NNUL_V;
                                        orders_cp-
>o_all_local_v2 = NNUL_V;
                                        orders_cp-
>o_all_local_1 = o_all_local / 256;
                                        orders_cp-
>o_all_local_2 = o_all_local % 256;

                                        /* Neworder
record    :     : 96/09/09 fukui */
                                        no_o_id_1 =
no_o_id / T256;
                                        no_o_id_2 =
(no_o_id-(no_o_id_1*T256))/D256;
                                        no_o_id_3
=(no_o_id-(no_o_id_1*T256)-
(no_o_id_2*D256))/256;
                                        no_o_id_4 =
no_o_id % T256;
                                        no_d_id_1 =
no_d_id / 256;
                                        no_d_id_2 =
no_d_id % 256;
                                        no_w_id_1 =
no_w_id / 256;
                                        no_w_id_2 =
no_w_id % 256;

                       fprintf(fst8 ,"%c%c%c%c%c"
                                       "%c%c%c%c%c",
                                        no_o_id_1,no_o_id_2,no_o
_id_3,no_o_id_4,
                                        no_d_id_1,no_d_id_2,no_w
_id_1,no_w_id_2);

                                 } else {
                                        /* ocarrier_id
*/
                                        o_carrier_id =
random_number(1, 10);

                                        orders_cp-
>o_carrier_id_v1 = NNUL_V;
                                        orders_cp-
>o_carrier_id_v2 = NNUL_V;
                                        orders_cp-
>o_carrier_id_1 = o_carrier_id / 256;
                                        orders_cp-
>o_carrier_id_2 = o_carrier_id % 256;

                                        /* order
record    :     : 96/09/09 fukui */
```

```
                        orders_cp->o_id_v1
= NNUL_V;
                        orders_cp->o_id_v2
= NNUL_V;
                        orders_cp->o_id_1
= o_id / T256;
                        orders_cp->o_id_2
= (o_id-(o_id_1*T256))/D256;
                        orders_cp->o_id_3
= (o_id-(o_id_1*T256)-(o_id_2*D256))/256;
                        orders_cp->o_id_4
= o_id % T256;

                        orders_cp-
>o_d_id_v1 = NNUL_V;
                        orders_cp-
>o_d_id_v2 = NNUL_V;
                        orders_cp-
>o_d_id_1 = o_d_id / 256;
                        orders_cp-
>o_d_id_2 = o_d_id % 256;

                        orders_cp-
>o_w_id_v1 = NNUL_V;
                        orders_cp-
>o_w_id_v2 = NNUL_V;
                        orders_cp-
>o_w_id_1 = o_w_id / 256;
                        orders_cp-
>o_w_id_2 = o_w_id % 256;

            /* 97-02-18 o_c_id          short->int  */
                        orders_cp-
>o_c_id_v1 = NNUL_V;
                        orders_cp-
>o_c_id_v2 = NNUL_V;
                        orders_cp-
>o_c_id_1 = o_c_id / T256;
                        orders_cp-
>o_c_id_2 = (o_c_id-(o_c_id_1*T256)) / D256;
                        orders_cp-
>o_c_id_3 = (o_c_id-(o_c_id_1*T256)
                                -
(o_c_id_2*D256))/256;
                        orders_cp-
>o_c_id_4 = o_c_id % T256;

                        orders_cp-
>o_ol_cnt_v1 = NNUL_V;
                        orders_cp-
>o_ol_cnt_v2 = NNUL_V;
                        orders_cp-
>o_ol_cnt_1 = o_ol_cnt / 256;
                        orders_cp-
>o_ol_cnt_2 = o_ol_cnt % 256;

                        orders_cp-
>o_all_local_v1 = NNUL_V;
                        orders_cp-
>o_all_local_v2 = NNUL_V;
                        orders_cp-
>o_all_local_1 = o_all_local / 256;
                        orders_cp-
>o_all_local_2 = o_all_local % 256;

                    }

          orders_cp    = orders_cp    + 1 ;
          orders_lpcnt = orders_lpcnt + 1 ;
          if ( orders_lpcnt == ORDERS_COUNT )
           {
            write(op_orders ,
                orders_ap ,
                  (size_t)ORDERS_SIZE *
(size_t)ORDERS_COUNT) ;
              orders_cp    = (orders_str
*)orders_ap   ;
              orders_lpcnt = 0          ;
            }

                        /*
ol_number                ,o_olcnt    */
                        /* order_line
*/
                        for (ol_number = 1;
ol_number <= o_ol_cnt; ol_number++)
            {

                            /* ol_i_id,
ol_dist_info          */
                            /* ol_i_id =
random_number(1, MAXITEMS); */

              /*    :961127:K.Fukui: I_ID
(main     ) */

                            ol_i_id =
random_number( 1, MAXITEMS / I_ID_Rand_by );
                            ol_i_id = ol_i_id *
I_ID_Rand_by;


      make_alpha_string(24, 24, ol_dist_info);
                            orderline_cp-
>ol_dist_info_v1 = NNUL_V;
                            orderline_cp-
>ol_dist_info_v2 = NNUL_V;
                strncpy(orderline_cp-
>ol_dist_info,ol_dist_info,24) ;

                            /* o_id>2100     */
                            /* ol_amount     */
                            /*           ,       */
                            if (o_id >
(CUST_PER_DIST - NEWORDS))
              {
                              /*   ol_amount /=
d_100;  */
                                ol_amount =
random_number(1, 999999);

                                /* orderline
*/

                              /*       ol_delivery_d
= NULL;*/
                                orderline_cp-
>ol_delivery_d_v1 = NUL_V;
                                orderline_cp-
>ol_delivery_d_v2 = NUL_V;

                                orderline_cp-
>ol_o_id_v1 = NNUL_V;
                                orderline_cp-
>ol_o_id_v2 = NNUL_V;
                        orderline_cp-
>ol_o_id_1 = ol_o_id / T256;
                        orderline_cp-
>ol_o_id_2 = (ol_o_id-(ol_o_id_1*T256))
                                        /D25
6;
                        orderline_cp-
>ol_o_id_3 = (ol_o_id-(ol_o_id_1*T256)
                                -
(ol_o_id_2*D256))/256;
                        orderline_cp-
>ol_o_id_4 = ol_o_id % T256;

                        orderline_cp-
>ol_d_id_v1 = NNUL_V;
                        orderline_cp-
>ol_d_id_v2 = NNUL_V;
                        orderline_cp-
>ol_d_id_1 = ol_d_id / 256;
                        orderline_cp-
>ol_d_id_2 = ol_d_id % 256;

                        orderline_cp-
>ol_w_id_v1 = NNUL_V;
                        orderline_cp-
>ol_w_id_v2 = NNUL_V;
                        orderline_cp-
>ol_w_id_1 = ol_w_id / 256;
                        orderline_cp-
>ol_w_id_2 = ol_w_id % 256;

                        orderline_cp-
>ol_number_v1 = NNUL_V;
                        orderline_cp-
>ol_number_v2 = NNUL_V;
                        orderline_cp-
>ol_number_1 = ol_number / 256;
                        orderline_cp-
>ol_number_2 = ol_number % 256;

                        orderline_cp-
>ol_i_id_v1 = NNUL_V;
                        orderline_cp-
>ol_i_id_v2 = NNUL_V;
                        orderline_cp-
>ol_i_id_1 = ol_i_id / T256;
                        orderline_cp-
>ol_i_id_2 = (ol_i_id-(ol_i_id_1*T256))
                                        /D25
6;
                        orderline_cp-
>ol_i_id_3 = (ol_i_id-(ol_i_id_1*T256)
                                -
(ol_i_id_2*D256))/256;
                        orderline_cp-
>ol_i_id_4 = ol_i_id % T256;

                        orderline_cp-
>ol_supply_w_id_v1 = NNUL_V;
                        orderline_cp-
>ol_supply_w_id_v2 = NNUL_V;
                        orderline_cp-
>ol_supply_w_id_1 = ol_supply_w_id / 256;
                        orderline_cp-
>ol_supply_w_id_2 = ol_supply_w_id % 256;

                        orderline_cp-
>ol_quantity_v1 = NNUL_V;
```

```
                                        orderline_cp-
>ol_quantity_v2 = NNUL_V;
                                        orderline_cp-
>ol_quantity_1 = ol_quantity / 256;
                                        orderline_cp-
>ol_quantity_2 = ol_quantity % 256;

                                        orderline_cp-
>ol_amount_v1 = NNUL_V;
                                        orderline_cp-
>ol_amount_v2 = NNUL_V;
                                        orderline_cp-
>ol_amount_1 = ol_amount / T256;
                                        orderline_cp-
>ol_amount_2 = (ol_amount
                                    -
(ol_amount_1*T256))/D256;
                                        orderline_cp-
>ol_amount_3 = (ol_amount
                                    -
(ol_amount_1*T256)
                                    -
(ol_amount_2*D256))/256;
                                        orderline_cp-
>ol_amount_4 = ol_amount % T256;

                                } else {
                                        ol_amount =
0;


        strcpy(ol_delivery_d, yyyymmddhhmmss);
                                        orderline_cp-
>ol_delivery_d_v1 = NNUL_V;
                                        orderline_cp-
>ol_delivery_d_v2 = NNUL_V;
                    strncpy(orderline_cp-
>ol_delivery_d,
                                ol_delivery_d,14);

                                        orderline_cp-
>ol_o_id_v1 = NNUL_V;
                                        orderline_cp-
>ol_o_id_v2 = NNUL_V;
                                        orderline_cp-
>ol_o_id_1 = ol_o_id / T256;
                                        orderline_cp-
>ol_o_id_2 = (ol_o_id
                                    -
(ol_o_id_1*T256))/D256;
                                        orderline_cp-
>ol_o_id_3 = (ol_o_id
                                    -
(ol_o_id_1*T256)
                                    -
(ol_o_id_2*D256))/256;
                                        orderline_cp-
>ol_o_id_4 = ol_o_id % T256;

                                        orderline_cp-
>ol_d_id_v1 = NNUL_V;
                                        orderline_cp-
>ol_d_id_v2 = NNUL_V;
                                        orderline_cp-
>ol_d_id_1 = ol_d_id / 256;
                                        orderline_cp-
>ol_d_id_2 = ol_d_id % 256;

                                        orderline_cp-
>ol_w_id_v1 = NNUL_V;
                                        orderline_cp-
>ol_w_id_v2 = NNUL_V;
                                        orderline_cp-
>ol_w_id_1 = ol_w_id / 256;
                                        orderline_cp-
>ol_w_id_2 = ol_w_id % 256;

                                        orderline_cp-
>ol_number_v1 = NNUL_V;
                                        orderline_cp-
>ol_number_v2 = NNUL_V;
                                        orderline_cp-
>ol_number_1 = ol_number / 256;
                                        orderline_cp-
>ol_number_2 = ol_number % 256;

                                        orderline_cp-
>ol_i_id_v1 = NNUL_V;
                                        orderline_cp-
>ol_i_id_v2 = NNUL_V;
                                        orderline_cp-
>ol_i_id_1 = ol_i_id / T256;
                                        orderline_cp-
>ol_i_id_2 = (ol_i_id
                                    -
(ol_i_id_1*T256))/D256;
                                        orderline_cp-
>ol_i_id_3 = (ol_i_id
                                    -
(ol_i_id_1*T256)
                                    -
(ol_i_id_2*D256))/256;
                                        orderline_cp-
>ol_i_id_4 = ol_i_id % T256;

                                        orderline_cp-
>ol_supply_w_id_v1 = NNUL_V;
                                        orderline_cp-
>ol_supply_w_id_v2 = NNUL_V;
                                        orderline_cp-
>ol_supply_w_id_1 = ol_supply_w_id / 256;
                                        orderline_cp-
>ol_supply_w_id_2 = ol_supply_w_id % 256;

                                        orderline_cp-
>ol_quantity_v1 = NNUL_V;
                                        orderline_cp-
>ol_quantity_v2 = NNUL_V;
                                        orderline_cp-
>ol_quantity_1 = ol_quantity / 256;
                                        orderline_cp-
>ol_quantity_2 = ol_quantity % 256;

                                        orderline_cp-
>ol_amount_v1 = NNUL_V;
                                        orderline_cp-
>ol_amount_v2 = NNUL_V;
                                        orderline_cp-
>ol_amount_1 = ol_amount / T256;
                                        orderline_cp-
>ol_amount_2 = (ol_amount
                                    -
(ol_amount_1*T256))/D256;
                                        orderline_cp-
>ol_amount_3 = (ol_amount
                                    -
(ol_amount_1*T256)
                                    -
(ol_amount_2*D256))/256;
                                        orderline_cp-
>ol_amount_4 = ol_amount % T256;

                                }

            orderline_cp    = orderline_cp    + 1 ;
            orderline_lpcnt = orderline_lpcnt + 1 ;
            if ( orderline_lpcnt ==
ORDERLINE_COUNT )
             {
               write(op_orderline ,
                  orderline_ap ,
                  (size_t)ORDERLINE_SIZE *
(size_t)ORDERLINE_COUNT);
               orderline_cp    = (orderline_str
*)orderline_ap ;
               orderline_lpcnt = 0           ;
             }
                        }
                    }
            }
            filecount++;
      }

     if ( orders_lpcnt != 0 )
      {
        write(op_orders ,
            orders_ap ,
            (size_t)ORDERS_SIZE *
(size_t)orders_lpcnt) ;
      }
     if ( orderline_lpcnt != 0 )
      {
        write(op_orderline ,
           orderline_ap ,
             (size_t)ORDERLINE_SIZE *
(size_t)orderline_lpcnt) ;
      }

      /*                */
      fclose(fst8);
      close(op_orders)    ;
      close(op_orderline) ;

  /*                */
  free(orders_ap)     ;

      /*                */
      return;

}

/*
 *     Function     : make_address()
 *     Description :                    .
 *     Parameters : 1. str1,     1(        21    )
 *                              2. str2,     2(        21    )
 *                              3. city,      (        21    )
 *                              4. state,
(        3    )
 *                              5. zip,
(        10    )
 *     Grobals Ref: nothing
 *     Grobals Out        : nothing
```

```
 *      Returns              : nothing
 */

void
make_address(str1, str2, city, state, zip)
char  *str1;
char  *str2;
char  *city;
char  *state;
char  *zip;
{
        /* street1       , 10-20                    */
        make_alpha_string(10, 20, str1);

        /* street2       , 10-20                    */
        make_alpha_string(10, 20, str2);

        /* city          , 10-20                    */
        make_alpha_string(10, 20, city);

        /* state         , 2                        */
        make_alpha_string(2, 2, state);

        /* zip           , 9                        */
        make_number_string(9, 9, zip);

        return;
}

/*
 *      Function     : lastname()
 *      Description : lastname                .
 *      Parameters : 1. num, 000-999
 *                   2. name,
 *      Grobals Ref: nothing
 *      Grobals Out        : nothing
 *      Returns            : nothing
 */

void
lastname(num, name)
int     num;
char  *name;
{
        /*    syllable  c_last                  10
*/
        static char  *syllable[] = {
                        "BAR", "OUGHT", "ABLE",
"PRI", "PRES",
                        "ESE", "ANTI", "CALLY",
"ATION", "EING"
                        };

        /* syllable[         100       ]  name          */
        strcpy(name, syllable[num / 100]);

        /* syllable[         10        ]  name          */
        strcat(name, syllable[(num /10) % 10]);

        /* syllable[         1         ]  name          */
        strcat(name, syllable[num % 10]);

        return;
}

/*
 *      Function     : make_alpha_string()
 *      Description :                .
 *      Parameters : 1. num1,
 *                   2. num2,
 *                   3. str,
 *      Grobals Ref: nothing
 *      Grobals Out        : nothing
 *      Returns            : int,
 */

int
make_alpha_string(num1, num2, str)
int     num1;
int     num2;
char  *str;
{
        int             len;
        int             i;
        short  rnum;

        /*                          num1-num2
*/
    if (num1 == num2) {
                len = num1;
        } else {
                len = random_number(num1, num2);
        }
        /*                                      */
        for ( i = 0; i < len; i++) {
#ifdef rand_str
                /* 0-61                 */
                rnum = random_number(0, 61);

                /* 0-25     ,                   'a'==x61 (0:a,
1:b, .. , 25:z) */
                if ((0 <= rnum) && (rnum <= 25)) {
                        str[i] = 'a' + rnum;

                /* 26-51      ,                  'A'==x41
(26:A, 27:B, .. , 51:Z) */
                } else if ((26 <= rnum) && (rnum <=
51)) {
                        str[i] = 'A' + rnum - 26;

                /* 52-61      ,                  '0'==x30 (52:0,
53:1, .. , 61:9) */
                } else if ((52 <= rnum) && (rnum <=
61)) {
                        str[i] = '0' + rnum - 52;
                }
#else
                /* 0-51                 */
        rnum = rand()%52  ;          /* 1997.03.11   */

                /* 0-25     ,                   'a'==x61 (0:a,
1:b, .. , 25:z) */
                if ((0 <= rnum) && (rnum <= 25)) {
                        str[i] = 'a' + rnum;

                /* 26-51      ,                  'A'==x41
(26:A, 27:B, .. , 51:Z) */
                } else if ((26 <= rnum) && (rnum <=
51)) {
                        str[i] = 'A' + rnum - 26;


                }
#endif
        }
        /*                  */
/*      if (num1 != num2) {*/
                str[len] = '\0';
/*      }*/

        /*                                      */
        return(len);
}

/*
 *      Function     : make_number_string()
 *      Description :                .
 *      Parameters : 1. num1,
 *                   2. num2,
 *                   3. str,
 *      Grobals Ref: nothing
 *      Grobals Out        : nothing
 *      Returns            : int,
 */

int
make_number_string(num1, num2, str)
int     num1;
int     num2;
char  *str;
{
        int             len;
        int             i;
        short  rnum;

        /*                          num1-num2
*/
        if (num1 == num2) {
                len = num1;
        } else {
                len = random_number(num1, num2);
        }
        /*                                      */
        for ( i = 0; i < len; i++) {

#ifdef rand_str
                /* 0-9                  */
                rnum = random_number(0, 9);

                /* 0-9                  str         */
                str[i] = '0' + rnum;
#else
       str[i] = (char)(rand()%10+'0');

#endif
        }

        /*                  */
        str[len] = '\0';

        /*                                      */
        return(len);
}

/*
 *      Function     : random_number()
 *      Description :                .
 *      Parameters : 1. num1,
 *                   2. num2,
 *      Grobals Ref: nothing
 *      Grobals Out        : nothing
 *      Returns            : int,
 */
#ifdef call_rand
/* 1997-02-18 mac           */
```

```
int
random_number(num1, num2)
int     num1;
int     num2;
{
        int     value;

        /* num1-num2                    */
        value = lrand48() % (num2 - num1 + 1) +
num1;

        /*                              */
        return(value);
}
#endif
/*
 *      Function     : set_seed()
 *      Description :                   .
 *      Parameters : 1. seedval,
 *      Grobals Ref: nothing
 *      Grobals Out         : nothing
 *      Returns             : nothing
 */

void
set_seed(seedval)
int     seedval;
{
        /*              */
        srand(seedval)  ;                       /* 1997-
02-18          */
        srand48(seedval);

        return;
}

/*
 *      Function     : nurand()
 *      Description :                       .
 *      Parameters : 1. a,
 *                   2. x,
 *                   3. y,
 *      Grobals Ref: nothing
 *      Grobals Out         : nothing
 *      Returns             : nothing
 */
/* 1997-02-18 TAB ID 221(c_last NURand     C)
*/
int
nurand(a, x, y, c)
int     a;
int     x;
int     y;
int c;
{
        int     value;

        /*                      */
        value = (((random_number(0, a) |
random_number(x, y)) + c) %
                (y - x + 1)) + x;

        /*                              */
        return(value);
}

/*
 *      Function     : init_permutation()
 *      Description : o_c_id    1  CUST_PER_DIST
 *                          .
 *      Parameters : nothing
 *      Grobals Ref: nothing
 *      Grobals Out         : 1. ocid, o_c_id
 *                    2. counter,
 *      Returns             : nothing
 */

void
init_permutation()
{
        short  cnt;
        short  replace;
        short  work;

        /*              ocid  1-CUST_PER_DIST
*/
        for (cnt = 0; cnt < CUST_PER_DIST;
cnt++){
                ocid[cnt] = cnt + 1;
        }

        /* ocid                         */
        for (cnt = 0; cnt < CUST_PER_DIST;
cnt++){
                replace = random_number(1,
CUST_PER_DIST);
                work = ocid[cnt];
                ocid[cnt] = ocid[replace - 1];
                ocid[replace -1] = work;
        }

        /*                          */
        counter = 0;
}
```

Appendix F: 180 Day Space Calculation

Warehouses	2772	tpmC	33113.60

Table	Rows	Data	Index	5% Space	8H Space	Total Space
Warehouse	2772	2801	0	140.1		2941.1
District	27720	27804	0	1390.2		29194.2
Item	100000	14288	0	714.4		15002.4
New-order	24948000	2530320	747936	163912.8		3442168.8
History	83160000	6523776	0		1246901.6	7770677.6
Orders	83160000	8340444	2116908		1998733.5	12456085.5
Customer	83160000	66529260	2027536	3427839.8		71984635.8
Order-line	831639940	89690112	0		17142640.7	106832752.7
Stock	277200000	100800420	0	5040021.0		105840441.0
DIRECTORY FILE		38912				38912.0
Dictionary		52080				52080.0
Totals		274550217	4892380	8634018.3	20388275.7	308464891.0

DB SPACE	318284592.00
Dynamic space	104554332.00
Static space	183522283.25
Free space	30207976.75
Daily growth	19983667.12
Daily spread	232476.07
180 day (KB)	3822428.06
180 day (GB)	3.645.35
8 Hour Log (GB)	96.61
After image Log [GB] (mirror)	8.52
Before image Log [GB] (mirror)	2.00
Log index [GB] (mirror)	0.20

DISKS PRICED		
Type	Number	Total GB
9GB DISK(LVS)	425	3604.70
4GB DISK	8	34.01
9GB DISK	22	187.42
Totals	480	3826.13

Space Usage	
Usage	Size (GB)
180-day Space	3645.35
Root,swap,usr	8.44
Log	161.30
Total	3815.10

Appendix G: Distribution of Tables and Log

PCI-BUS	Device	Capacity	Using	Filename	Size (MB)	Comment
#0	c0t0d0	4.3GB	Operating-system	/dev/rdsk/c0t0d0	------	
			swap		244	
	c0t1d0	8.5GB	AI-LOG	/dev/sfdsk/tpcc/rdsk/vo1_1	8512	mirrored by Safe-DISK
#1	c2t4d3	42.4GB	ARC-LOG	/dev/rdsk/c2t4d3	43426	5DISK(9GB-DISK*5),RAID0
	c2t4d4	42.4GB	ARC-LOG	/dev/rdsk/c2t4d4	43426	5DISK(9GB-DISK*5),RAID0
	c2t5d0	42.4GB	Striping D	/dev/rdsk/c2t5d0	4678	5DISK(9GB-DISK*5),RAID0
			swap	/dev/dsk/c2t5d0s7	2000	
	c2t5d1	42.4GB	Striping D	/dev/rdsk/c2t5d1	4678	5DISK(9GB-DISK*5),RAID0
	c2t5d2	42.4GB	Striping D	/dev/rdsk/c2t5d2	4678	5DISK(9GB-DISK*5),RAID0
	c3t4d3	42.4GB	DIRECTORY FILE	/rdbptc/RDBD2/DIR_FILE1,DIR_FILE2	50	5DISK(9GB-DISK*5),RAID0
			ARC-LOG	/dev/rdsk/c3t4d3	43376	5DISK(9GB-DISK*5),RAID0
	c3t4d4	42.4GB	Striping G	/dev/rdsk/c3t4d4	7778	5DISK(9GB-DISK*5),RAID0
	c3t5d0	42.4GB	Striping D	/dev/rdsk/c3t5d0	4678	5DISK(9GB-DISK*5),RAID0
			swap	/dev/dsk/c3t4d4s7	2048	
	c3t5d1	42.4GB	Striping D	/dev/rdsk/c3t5d1	4678	5DISK(9GB-DISK*5),RAID0
	c3t5d2	42.4GB	Striping E	/dev/rdsk/c3t5d2	6257	5DISK(9GB-DISK*5),RAID0
#2	c4t1d0	8.5GB	Striping H	/dev/rdsk/c4t1d0	4987	
	c4t2d0	8.5GB	Striping L	/dev/rdsk/c4t2d0	3629	
	c4t3d0	8.5GB	Striping I	/dev/rdsk/c4t3d0	4534	
	c4t4d0	8.5GB	Striping I	/dev/rdsk/c4t4d0	4534	
	c4t5d0	8.5GB	Striping I	/dev/rdsk/c4t5d0	4534	
	c5t1d0	8.5GB	BI-LOG	/dev/sfdsk/tpcc/rdsk/vo2_1	2048	mirrored by Safe-DISK
			IX-LOG	/dev/sfdsk/tpcc/rdsk/vo2_2	208	mirrored by Safe-DISK
	c5t2d0	8.5GB	Striping I	/dev/rdsk/c5t2d0	4534	
	c5t3d0	8.5GB	Striping J	/dev/rdsk/c5t3d0	4531	
	c5t4d0	8.5GB	Striping J	/dev/rdsk/c5t4d0	4531	
	c5t5d0	8.5GB	Striping J	/dev/rdsk/c5t5d0	4531	
#3	c6t4d3	42.4GB	Striping E	/dev/rdsk/c6t4d3	6257	5DISK(9GB-DISK*5),RAID0
	c6t4d4	42.4GB	Striping E	/dev/rdsk/c6t4d4	6257	5DISK(9GB-DISK*5),RAID0
	c6t5d0	42.4GB	Striping A	/dev/rdsk/c6t5d0	4683	5DISK(9GB-DISK*5),RAID0
	c6t5d1	42.4GB	Striping B	/dev/rdsk/c6t5d1	4683	5DISK(9GB-DISK*5),RAID0
	c6t5d2	42.4GB	Striping B	/dev/rdsk/c6t5d2	4683	5DISK(9GB-DISK*5),RAID0
	c7t4d3	42.4GB	Striping E	/dev/rdsk/c7t4d3	6257	5DISK(9GB-DISK*5),RAID0
	c7t4d4	42.4GB	Striping E	/dev/rdsk/c7t4d4	6257	5DISK(9GB-DISK*5),RAID0
	c7t5d0	42.4GB	Striping D	/dev/rdsk/c7t5d0	4678	5DISK(9GB-DISK*5),RAID0
	c7t5d1	42.4GB	Striping D	/dev/rdsk/c7t5d1	4678	5DISK(9GB-DISK*5),RAID0
	c7t5d2	42.4GB	Striping D	/dev/rdsk/c7t5d2	4678	5DISK(9GB-DISK*5),RAID0

PCI-BUS	Device	Capacity	Using	Filename	Size (MB)	Comment
#4	c8t1d0	4.3GB	Striping K	/dev/rdsk/c8t1d0	3625	
	c8t2d0	4.3GB	Striping K	/dev/rdsk/c8t2d0	3625	
	c8t3d0	4.3GB	Striping K	/dev/rdsk/c8t3d0	3625	
	c8t4d0	4.3GB	Striping K	/dev/rdsk/c8t4d0	3625	
	c8t5d0	4.3GB	Striping K	/dev/rdsk/c8t5d0	3625	
	c9t1d0	8.5GB	AI-LOG	/dev/sfdsk/tpcc/rdsk/vo1_1	8512	mirrored by Safe-DISK
	c9t2d0	8.5GB	BI-LOG	/dev/sfdsk/tpcc/rdsk/vo2_1	2048	mirrored by Safe-DISK
			IX-LOG	/dev/sfdsk/tpcc/rdsk/vo2_2	208	mirrored by Safe-DISK
	c9t3d0	4.3GB	Striping M	/dev/rdsk/c9t3d0	2717	
	c9t4d0	4.3GB	Striping K	/dev/rdsk/c9t4d0	3625	
	c9t5d0	4.3GB	Striping K	/dev/rdsk/c9t5d0	3625	
#5	c10t4d3	42.4GB	Striping F	/dev/rdsk/c10t4d3	6199	5DISK(9GB-DISK*5),RAID0
	c10t4d4	42.4GB	Striping E	/dev/rdsk/c10t4d4	6257	5DISK(9GB-DISK*5),RAID0
	c10t5d0	42.4GB	Striping D	/dev/rdsk/c10t5d0	4678	5DISK(9GB-DISK*5),RAID0
	c10t5d1	42.4GB	Striping D	/dev/rdsk/c10t5d1	4678	5DISK(9GB-DISK*5),RAID0
	c10t5d2	42.4GB	Striping D	/dev/rdsk/c10t5d2	4678	5DISK(9GB-DISK*5),RAID0
	c11t4d3	42.4GB	Striping F	/dev/rdsk/c11t4d4	6199	5DISK(9GB-DISK*5),RAID0
	c11t4d4	42.4GB	Striping E	/dev/rdsk/c11t4d4	6257	5DISK(9GB-DISK*5),RAID0
	c11t5d0	42.4GB	Striping C	/dev/rdsk/c11t5d0	4678	5DISK(9GB-DISK*5),RAID0
	c11t5d1	42.4GB	Striping D	/dev/rdsk/c11t5d1	4678	5DISK(9GB-DISK*5),RAID0
	c11t5d2	42.4GB	Striping D	/dev/rdsk/c11t5d2	4678	5DISK(9GB-DISK*5),RAID0
#6	c12t1d0	8.5GB	Striping H	/dev/rdsk/c12t1d0	4987	
			DICTIONARY	/dev/rdsk/c12t1d0s5	101	DICTIONARY
	c12t2d0	8.5GB	Striping L	/dev/rdsk/c12t2d0	3629	
	c12t3d0	8.5GB	Striping I	/dev/rdsk/c12t3d0	4534	
	c12t4d0	8.5GB	Striping I	/dev/rdsk/c12t4d0	4534	
	c12t5d0	8.5GB	Striping I	/dev/rdsk/c12t5d0	4534	
	c13t1d0	8.5GB	Striping H	/dev/rdsk/c13t1d0	4987	
	c13t2d0	8.5GB	Striping I	/dev/rdsk/c13t2d0	4534	
	c13t3d0	8.5GB	Striping I	/dev/rdsk/c13t3d0	4534	
	c13t4d0	8.5GB	Striping I	/dev/rdsk/c13t4d0	4534	
	c13t5d0	8.5GB	Striping I	/dev/rdsk/c13t5d0	4534	
#7	c14t4d3	42.4GB	Striping E	/dev/rdsk/c14t4d3	6257	5DISK(9GB-DISK*5),RAID0
	c14t4d4	42.4GB	Striping E	/dev/rdsk/c14t4d4	6257	5DISK(9GB-DISK*5),RAID0
	c14t5d0	42.4GB	Striping D	/dev/rdsk/c14t5d0	4678	5DISK(9GB-DISK*5),RAID0
	c14t5d1	42.4GB	Striping D	/dev/rdsk/c14t5d1	4678	5DISK(9GB-DISK*5),RAID0
	c14t5d2	42.4GB	Striping D	/dev/rdsk/c14t5d2	4678	5DISK(9GB-DISK*5),RAID0
	c15t4d3	42.4GB	Striping E	/dev/rdsk/c15t4d4	6257	5DISK(9GB-DISK*5),RAID0
	c15t4d4	42.4GB	Striping F	/dev/rdsk/c15t4d4	6199	5DISK(9GB-DISK*5),RAID0
	c15t5d0	42.4GB	Striping D	/dev/rdsk/c15t5d0	4678	5DISK(9GB-DISK*5),RAID0
	c15t5d1	42.4GB	Striping D	/dev/rdsk/c15t5d1	4678	5DISK(9GB-DISK*5),RAID0
	c15t5d2	42.4GB	Striping D	/dev/rdsk/c15t5d2	4678	5DISK(9GB-DISK*5),RAID0

Striping A		Striping B		Striping C	
Table name	Size (Kbyte)	Table name	Size (Kbyte)	Table name	Size (Kbyte)
Customer	1663488	Customer	1663488	Customer	1663488
Customer Index	51072	Customer Index	51072	Customer Index	51072
Orders	253824	Orders	253824	Orders	253824
Orders Index	101760	Orders Index	101760	Orders Index	101760
History	200448	History	200448	History	200448
Stock	2520064	Stock	2520064	Stock	2520064
Item	5184	Item	4224	Item	1408
Total	4795840	Total	4794880	Total	4792064

Striping D		Striping E		Striping F	
Table name	Size (Kbyte)	Table name	Size (Kbyte)	Table name	Size (Kbyte)
Customer	1663488	Customer	2217984	Customer	2217984
Customer Index	51072	Customer Index	68096	Customer Index	68096
Orders	253824	Orders	338432	Orders	338432
Orders Index	101760	Orders Index	135680	Orders Index	135680
History	200448	History	267264	History	267264
Stock	2520064	Stock	3380096	Stock	3320064
Total	4790656	TOTAL	6407552	TOTAL	6347520

Striping G		Striping H		Striping I	
Table name	Size (Kbyte)	Table name	Size (Kbyte)	Table name	Size (Kbyte)
Customer	2772480	WareHouse	4224	WareHouse	4224
Customer Index	85120	District	2688	District	2688
Orders	423040	NewOrder	163328	NewOrder	148480
Orders Index	169600	NewOrder Index	74624	NewOrder Index	67840
History	334080	OrderLine	4861824	OrderLine	4419840
Stock	4180096				
TOTAL	7964416	TOTAL	5106688	TOTAL	4643072

Striping K		Striping L		Striping M	
Table name	Size (Kbyte)	Table name	Size (Kbyte)	Table name	Size (Kbyte)
WareHouse	2112	WareHouse	4224	NewOrder	89088
District	1344	District	2688	NewOrder Index	40704
NewOrder	118784	NewOrder	118784	OrderLine	2651904
NewOrder Index	54272	NewOrder Index	54272		
OrderLine	3535872	OrderLine	3535872		
TOTAL	3712384	TOTAL	3715840	TOTAL	2781696

Appendix H: Price Quotes

BEA ENTERPRISE MIDDLEWARE SOLUTIONS

June 25, 1999

FUJITSU LIMITED
Tokyo, Japan

Gentlemen:

Per your request I am enclosing the pricing information regarding TUXEDO 6.x that you requested. This pricing applies to Tuxedo 6.1, 6.2, 6.3, 6.4 and 6.5. Please note that Tuxedo 6.5 is our most recent version of Tuxedo but that all 6.x releases are generally available. Core functionality services pricing is appropriate for your activities. As per the table below, Fujitsu server systems are classified in one of 5 tiers based on CPU type and capacity.

This pricing quotation will be valid through August 25, 1999.

Tuxedo Core Functionality Services (CFS) Program Product Pricing and Description

TUX-CFS provides a basic level of middleware support for distributed computing, and is best used by organizations with substantial resources and knowledge for advanced distributed computing implementations.

TUX-CFS prices are server only and are based on the overall performance characteristics of the server and uses the same five tier computer classification as TUXEDO 6.x. Prices range from $3,000 for Tier 1 to $250,000 for Tier 5. Under this pricing option EVERY system running TUX-CFS at the user site must have a TUXEDO license installed and pay the appropriate per server license fees.

BEA Tux/CFS Version 6.x Unlimited User License Fees Per Server

Unlimited User License fees per server	Number of Users	Dollar Amount	Maintenance (5 x 8) per year	Maintenance (7 x 24) per year
Tier 1 -- PC Servers with 1 or 2 CPUs, entry level RISC Uni-processor workstations and servers	Unlimited	$3,000.00	$450.00	$660.00
Tier 2 -- PC Servers with 3 or 4 CPUs, Midrange RISC Uni-processor servers and workstations	Unlimited	$12,000.00	$1,800.00	$2,640.00
Tier 3 – Midrange Multiprocessors, up to 8 CPUs per system capacity	Unlimited	$30,000.00	$4,500.00	$6,600.00
Tier 4 – Large (more than 8, less than 32 CPUs) and Mainframe Systems	Unlimited	$100,000.00	$15,000.00	$22,000.00
Tier 5 -- Massively Parallel Systems, > 32 processors; All Mainframes	Unlimited	$250,000.00	$37,500.00	$55,000.00

1

99- 6-24; 9:09 ;BEA Systems Japan 10452241267 # 2/ 3

Intel based server tier classifications:

Platform	Operating System	Tier 1	Tier 1	Tier 2	Tier 3	Tier 3
Intel Pentium/ Pentium Pro PCs	Interactive R3.2 ESIX SVR 4.0 SCO UNIX 3.2.2 and 3.2.4 SCO ODT 2.x,3.x Solaris x86 2.X UnixWare, Windows NT 3.5/4.0	All 386/486 PCs are Class 1	ALL Pentium and Pentium Pro PCs with 1 or 2 CPUs capacity are Tier 1	ALL Pentium and Pentium Pro PCs with 3 or 4 CPUs capacity are Tier 2		ALL Pentium and Pentium Pro PCs with 5,6,7, or 8 CPUs are Tier 3

Sun Microsystems Server Tier classifications

Tier 1	Tier 2	Tier 3	Tier 3	Tier 4	Tier 5
Class 2	Class 3	Class 4	Class 5	Class 6	Class 7
Station 5/85 Station 4 Station 20/50 Station 20/51 Station 20/61 Ultra 1 140/170 Server 470 Server 5/70 Server 20/50 Server 20/51 Server 20/61 Ultra 2 Desktop Ultra 5 Ultra 10, 10S	Server 5/85 Station 20/71 Server 20/71 Ultra Enterprise 1 &150 Ultra Enterprise 2 -2100,2200 Ultra 60	Station 20/502 MP 20/612 MP 20/514 MP 20/HS11 20/712 MP Server 1000 Server 1000E Server 20/502 Server 20/712 Server 20/612 Server 20/514 Ultra Enterprise 2 -2300 Ultra 450	SparcCenter 1000 Ultra Enterprise 3000 Ultra Enterprise 4000 & 5000 < 8 proc.	SparcCenter 2000 SparcCenter 2000E Ultra Enterprise 4000 & 5000 & 6000 Between 8 and 32 proc. CRS6400 (< 32 proc.)	CRS6400 (≥32 proc.) Ultra Enterprise 6000 (≥32 proc.) Ultra Enterprise 10000 (all systems)

Very Truly Yours,

Lewis D. Brentano,
Director, Market Planning

Appendix I: Auditor's attestation letter

INFO SIZING

TPC CERTIFIED AUDITOR

Sponsor: Mr. Kazuhiko Saito
Director, Development DEPT. III
ENTERPRISE MIDDLEWARE DIV.
SOFTWARE GROUP
Fujitsu Limited
140 Miyamtoto
Numazu-shi, Shizuoka, 410-0396, Japan

June 23, 1999

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

Platform: Fujitsu GP7000F Model 600 c/s
Operating system: Solaris Version 2.6
Database Manager: SymfoWARE Server Enterprise Edition for VLM 2.0
Transaction Manager: BEA Tuxedo Version 6.4 CFS

The results were:

CPU's Speed	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: Fujitsu GP7000F Model 600				
8 x SPARC64 GP (300 MHz)	8 GB Main 8 MB cache/cpu	223 x 9 GB (8+1) x 4 GB	1.60 Seconds	33,113.60
Six (6) Clients: Fujitsu GRANPOWER 5000 model 580 (Specification for each)				
2 x Pentium II Xeon (400 MHz)	512 MB Main 512 KB cache/cpu	1 x 4 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC requirements for Revision 3.4 of the benchmark. The following verification items were given special attention:

- The transactions were correctly implemented
- The database records were the proper size

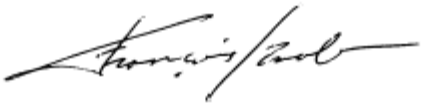
1373 North Franklin Street • Colorado Springs, CO 80903-2527 • Office: 719/473-7555 • Fax: 719/473-7554

- The database was properly scaled and populated
- The ACID properties were met
- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured.
- At least 90% of all delivery transactions met the 80 Second completion time limit
- All 90% response times were under the specified maximums
- The measurement interval was representative of steady state conditions
- The reported measurement interval was 30 minutes (1800 seconds)
- One checkpoints was taken during the measurement interval
- Measurement repeatability was verified
- The 180 day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

None.

Respectfully Yours,



François Raab
President