TPC Benchmark™C

Full Disclosure Report

Fujitsu
GP 7000
Model 600 c/s w/ 9 Front-Ends

running

SymfoWARE Server for VLM 2.0

July 30, 1998

The benchmark results contained in this document were submitted for compliance with version 3.3.3 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 July 30, 1998.

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.

Copyright 1998 Fujitsu

Printed in the United States July 30, 1998

Solaris 2.6 is a trademark of Sun Microsystems, Inc.

SymfoWARE is a trademark of Fujitsu in Japan.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/OPEN Company Limited.

SPARC is a registered trademark of SPARC International.

Ultra is a trademark of Sun Microsystems, Inc.

Microsoft, Windows, MS-DOS and the Microsoft logo are registered trademarks of Microsoft Corporation.

TUXEDO 4.2, is Copyright © 1996-1998 BEA Systems, Inc. Portions of this software © 1995 Novell, Inc. All rights reserved.

TPC Benchmark, TPC-C and tpmC are trademarks of the Transaction Processing Performance Council.

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 3.3.3, released April 16th, 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 GP 7000 Model 600 c/s w/ 9 Front-Ends. The operating system used for the benchmark was Solaris 2.6. The DBMS used was SymfoWARE Server 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:

20,170.63 tpmC
$110.30 per tpmC
January 21'st, 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 Tom Sawyer of Performance Metrics, Inc. to verify compliance with the relevant TPC specifications.

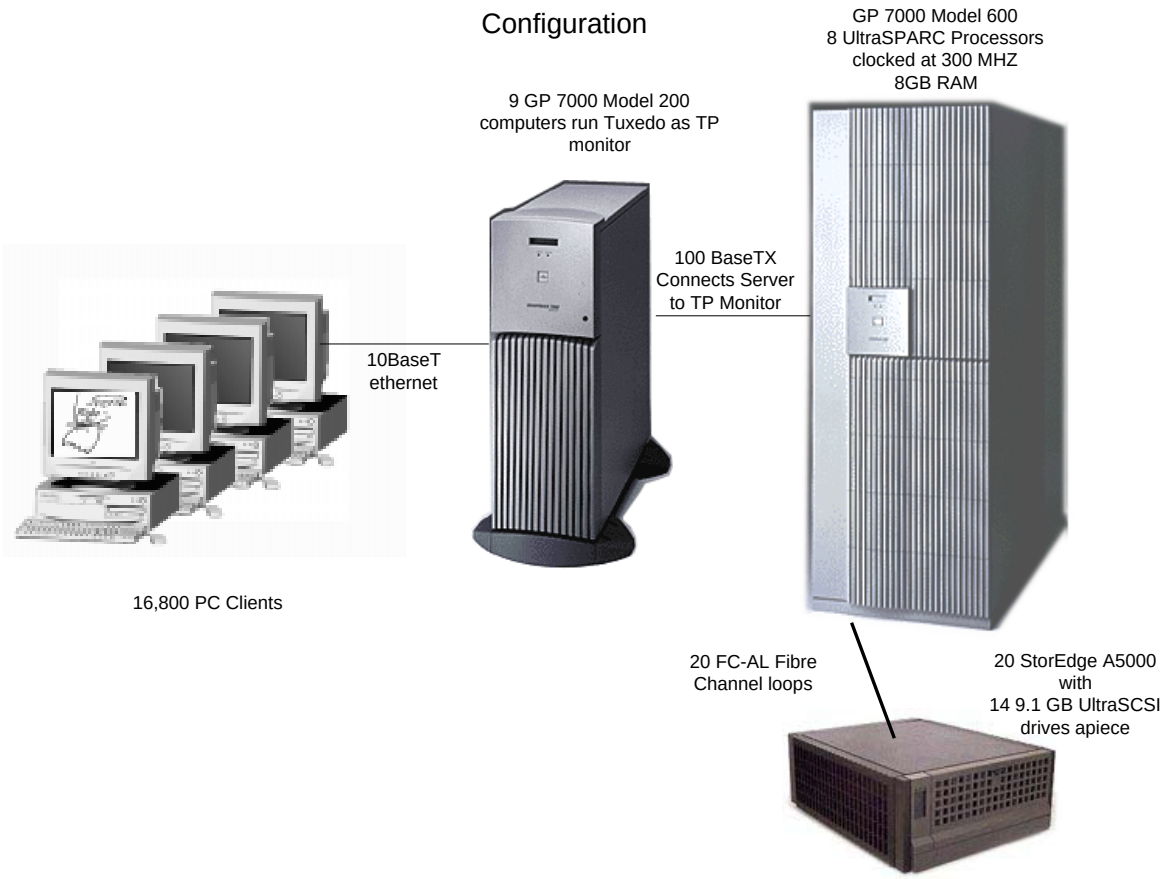
Priced Configuration

FUJITSU	GP 7000 Model 600 c/s w/ 9 Front-Ends	TPC-C Rev 3.3.3 Report Date: July 30, 1998

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
$2,224,842.20	20,170.63 tpmC	$110.30/tpmC	January 21'st, 1999

Processors	Database Manager	Operating system	Other Software	Number of users
8 UltraSPARC @300Mhz	SymfoWARE Server for VLM 2.0	Solaris 2.6	Tuxedo 4.2 Volume Manager 2.5 COBOL85 V20	16,800

GP 7000 Model 600 Priced Configuration



RDBMS SERVER			CLIENTS	
SYSTEM COMPONENTS	QTY	DESCRIPTION	QTY	DESCRIPTION
PROCESSOR	8	ULTRASPARC @ 300MHZ	9	1 ULTRASPARC @ 250MHZ
CACHE MEMORY		2MB (EACH PROCESSOR)		1 MB (EACH)
MEMORY		8GB	9	1GB
DISK CONTROLLER	10	FC-AL HOST ADAPTORS	--	--
DISKS	280	8.4 GB DISK	18	2.0GB DISK
	7	4.0 GB DISK		
	8	2.0 GB DISK		
TOTAL		2396GB		
SCSI CONTROLLER	3	FAST-WIDE SCSI		
TERMINAL	1	CONSOLE	1	CONSOLE
NETWORK INTERFACE	1	100 BASE-TX	3	2 10BASET, 1 100BASETX
HUBS	3	8-PORT (100 BASE-TX)	2,117	NETLUX (8 PORT HUBS)
PC'S			16,800	PC'S RUNNING WINDOWS95

Numerical Quantities Summary

GP 7000 Model 600 c/s w/ 9 Front-Ends

SymfoWARE Server for VLM 2.0

MQTH, Computed Maximum Qualified Throughput	**20,170.63 tpmC**

Response Times (in seconds)	**Average**	**90%**	**Max.**
New-Order	**1.59**	**1.63**	**84.58**
Payment	**1.09**	**1.43**	**89.27**
Order-Status	**0.94**	**1.42**	**79.16**
Delivery (interactive portion)	**0.31**	**0.22**	**62.12**
Delivery (deferred portion)	**1.56**	**2.20**	**77.86**
Stock-Level	**0.78**	**1.41**	**72.36**
Menu	**0.01**	**0.01**	**1.38**

Transaction Mix, in percent of total transaction	
New-Order	**44.75**
Payment	**43.11**
Order-Status	**4.05**
Delivery	**4.05**
Stock-Level	**4.04**

Emulation Delay (in seconds)	**Resp. Time**	**Menu**
New-Order	N/A	N/A
Payment	N/A	N/A
Order-Status	N/A	N/A
Delivery (interactive)	N/A	N/A
Stock-Level	N/A	N/A

Keying/Think Times (in seconds)	**Min.**		**Average**		**Max.**	
New-Order	18.09	0	18.11	12.06	18.36	120.33
Payment	3.04	0	3.06	12.06	3.31	120.17
Order-Status	2.04	0	2.05	10.15	2.22	96.18
Delivery (interactive)	2.04	0	2.06	5.07	2.25	48.37
Stock-Level	2.04	0	2.06	5.02	2.24	48.96

Test Duration	
Ramp-up time (seconds)	**4680**
Measurement interval	**1800**
Transactions during measurement interval	**605,119**
Ramp down time	

Checkpointing	
Number of checkpoints	**0**
Checkpoint interval	**1800**

Reproducibility Run	
Reported measurement	**20,170.63**
Reproducibility measurement	**20,126.73**
Difference	**43.90**

Detailed Pricing Information

FUJITSU

Detailed Pricing Information

GP 7000

Model 600 c/s w/ 9 Front-Ends

TPC-C Rev April 16th, 1998

Report Date:

July 30, 1998

Order Number	Description	Quantity	Third Party	Unit Price	Extended Price	Maintenance rate/unit*		5 Years Maintenance
Server Hardware								
882-00008-A	GP7000 Model 600 w/2 x UltraSPARC-II 300MHz/2MB cache	1		39,500.00	39,500.00		/5yr	0.00
822-00023	2 x UltraSPARC-II 300MHz/2MB cache (2 to 4)	1		20,500.00	20,500.00		/mo	0.00
822-00024	2 x UltraSPARC-II 300MHz/2MB cache (4 to 6)	1		20,500.00	20,500.00		/mo	0.00
822-00025	2 x UltraSPARC-II 300MHz/2MB cache (6 to 8)	1		20,500.00	20,500.00		/mo	0.00
952-00065	Software support	1		0.00	0.00	8,757.00	/yr	8,757.00
952-00066	Hardware + Software support	1		0.00	0.00	14,594.00	/yr	58,376.00
952-00070	Additional Processor support	6		0.00	0.00	450.00	/yr	10,800.00
822-00005	1GB (4 x 256MB) Memory	8		12,500.00	100,000.00		/mo	0.00
822-00011	19"Rackmount Disk Expansion Unit	2		2,450.00	4,900.00		/mo	0.00
512-00009	Internal HDD (2GB)	8		900.00	7,200.00		/mo	0.00
822-00006	Internal HDD (4GB)	7		1,250.00	8,750.00		/mo	0.00
822-00031	SBus slot upgrade (8 to 16)	1		6,500.00	6,500.00		/mo	0.00
820-00031	Fast/Wide/Differential SCSI-2 adapter	2		1,300.00	2,600.00		/mo	0.00
SG-XARY011A-127G	A5000 disk array w/ 14 x 9.1GB disks	20	1,2	37,800.00	756,000.00	9,127.76	/5yr	182,555.20
X6730A	FC-AL host adapter w/ 1 GBIC module	10	1,2	1,620.00	16,200.00		/mo	0.00
X6732A	FC-AL 7port hub unit	20	1,2	1,380.00	27,600.00		/mo	0.00
X978A	Fibre Optic Cable (15m)	20	1,2	175.00	3,500.00		/mo	0.00
X6731A	GBIC module	70	1,2	360.00	25,200.00		/mo	0.00
820-00027	FastEthernet Adapter (10/100Base-TX)	1		800.00	800.00		/mo	0.00
WY-55	Wyse WY-55 terminal + 2 year warranty extension	1	3	295.00	295.00	100.00	/2yr	100.00
822-00008	8mm Tape unit	1		1,360.00	1,360.00		/mo	0.00
	Server Hardware Subtotals				1,061,905.00			260,588.20
Server Software								
830-00062	Solaris 2.6 and Server Supplement Software	1		100.00	100.00		/yr	0.00
B23G0X1H	SymfoWARE Server for VLM 2.0	1		186,723.00	186,723.00	37,344.60	/5yr	186,723.00
D151DX0	C Workbench 2.0.3	1		995.00	995.00		/yr	0.00
	Server Software Subtotals				187,818.00			186,723.00
Client Hardware								
882-00002	GP7000 Model 200 w/ 1 x UltraSPARC-II 250MHz/1MB cache	9		9,500.00	85,500.00		/yr	0.00
952-00023	Software support	9		0.00	0.00	1,895.00	/yr	17,055.00
952-00024	Hardware + Software support	9		0.00	0.00	2,940.00	/yr	105,840.00
822-00003	256MB (4 x 64MB) Memory	36		2,800.00	100,800.00		/mo	0.00
512-00009	Internal HDD (2GB)	18		900.00	16,200.00		/mo	0.00
820-00027	FastEthernet Adapter (10/100Base-TX)	18		800.00	14,400.00		/mo	0.00
WY-55	Wyse WY-55 terminal + 2 year warranty extension	9	3	295.00	2,655.00	100.00	/2yr	900.00
	Client Hardware Subtotals				219,555.00			123,795.00
Client Software								
20-0301-2121-0401-200	Fujitsu COBOL Professional for Sun Solaris V4	1		2,000.00	2,000.00	500.00	/yr	2,500.00
	BEA Tuxedo 4.2	9	4	4,200.00	37,800.00	630.00	/yr	28,350.00
	Client Software Subtotals				39,800.00			30,850.00
User Connectivity								
NH2012	100BASE-TX Switching Hub units (8ports) *	5	3	2,100.00	10,500.00	0.00	/mo	0.00
NH2012-61A	5 year advance exchange warranty	5	3	0.00	0.00	1,098.00	/5yr	5,490.00
NX-H9+	UTP 8 Port Hub with BNC *	2329	5	42.00	97,818.00	0.00	/mo	0.00
	User Connectivity Subtotals				108,318.00			5,490.00
	Totals				1,617,396.00			607,446.20
	5 Year cost							2,224,842.20
	tpmC							20,170.63
	$ / tpmC							110.30

Third Party Pricing:

1 = Sun Microsystems
2 = SUNSERVICE (Suppor)
3 = SSP Data
4 = BEA Systems
5 = NETLUX

* : 10% or minimum of 2 spares are included

Notes:

- GP hardware has a 12 month warranty. Thus to cost 5 years of hardware maintenance, a total of 48 months is calculated.

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.

FUJITSU

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

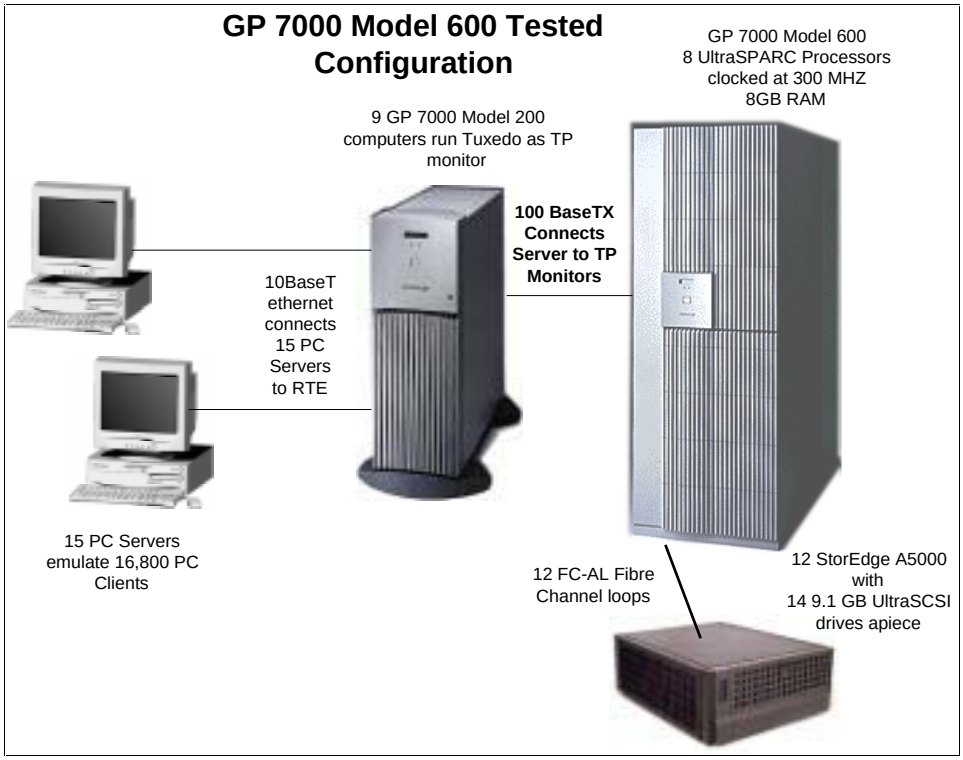
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

The System Under Test (SUT), a GP 7000 Model 600 c/s w/ 9 Front-Ends, is depicted in the following diagrams.

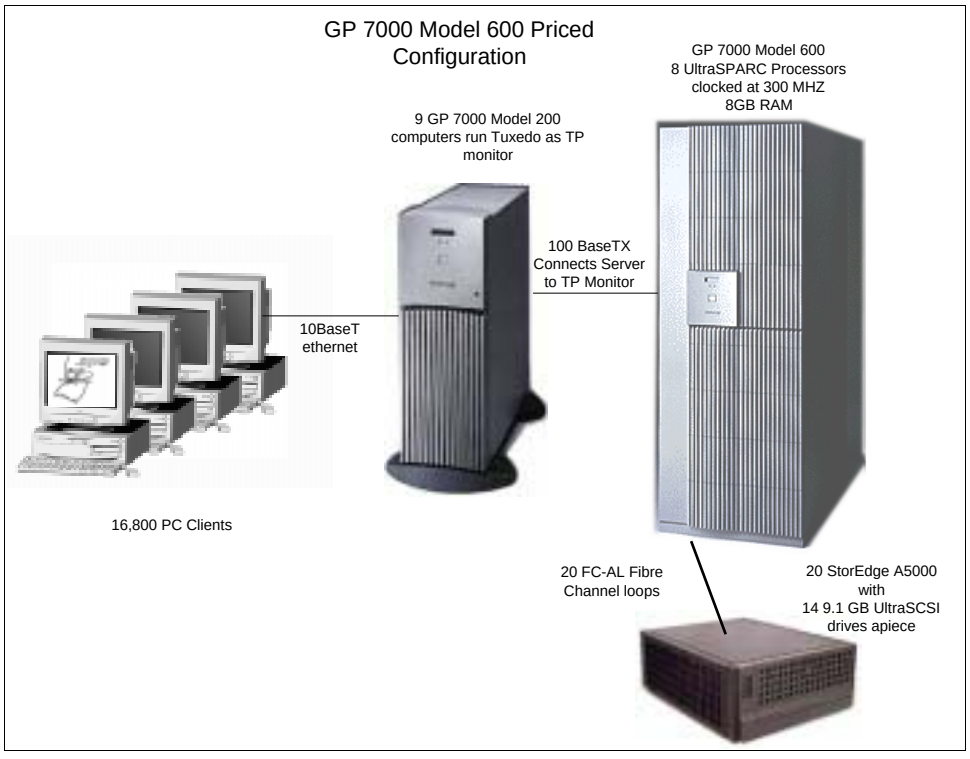
The configuration diagrams for both the tested and priced systems are included on the following pages.

The only difference is the use of the RTE.

GP 7000 Tested Configuration



GP 7000 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 F 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:

Warehouse	60
District	60
Customer	10
History	10
Order	10
New Order	10
OrderLine	10
Stock	30

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.

Presentation is handled by the terminal emulator bundled with Windows 95.

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	1.00%
	Average items per order	10
Payment	Home warehouse	84.89%
	Remote warehouse	15.11%
	Accessed by last name	60.03%
Order Status	Accessed by last name	60.18%
Delivery	Skipped transactions	None
Transaction Mix	New Order	44.75%
	Payment	43.11%
	Order status	4.05%
	Delivery	4.05%
	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 the fully-scaled database used for the performance measurements:

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 RTE's were started with 16,000 users.
4. The test was allowed to run for a minimum of five 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 diakarray cabinet.
7. The RTE's were shut down
8. A new disk was inserted into the diskarray cabinet and the data disks were 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 RTE's.
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 1,680 warehouses under a full load of 16,000 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 16,000 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 configured with 1,680 warehouses.

Table 4.1 Number of Rows for Server

Table	Occurrences
Warehouse	1,680
District	16,800
Customer	50,400,000
History	50,400,000
Order	50,400,000
New Order	15,120,000
Order Line	503,989,400
Stock	168,000,000
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 COBOL85 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 Section 1.4.

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 8.5184KB per New-Order transaction, which was measured from the steady state. The 8 hours log space was 78.65GB at the measured rate and 84.3GB 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 were 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	20,170.63
Price per tpmC	$110.30

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.

Table 5.1 Response Times

Type	Average	Maximum	90th %
New-Order	1.59	84.58	1.63
Payment	1.09	89.27	1.43
Order-Status	0.94	79.16	1.42
Interactive Delivery	0.31	62.12	0.22
Deferred Delivery	1.56	77.86	2.20
Stock-Level	0.78	72.36	1.41
Menu	0.01	1.38	0.01

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	18.09	18.11	18.36
Payment	3.04	3.06	3.31
Order-Status	2.04	2.05	2.22
Interactive Delivery	2.04	2.06	2.25
Stock-Level	2.04	2.06	2.24

Table 5.3 Think Times

Type	Minimum	Average	Maximum
New-Order	0.00	12.06	120.33
Payment	0.00	12.06	120.17
Order-Status	0.00	10.15	96.18
Interactive Delivery	0.00	5.07	48.37
Stock-Level	0.00	5.02	48.96

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

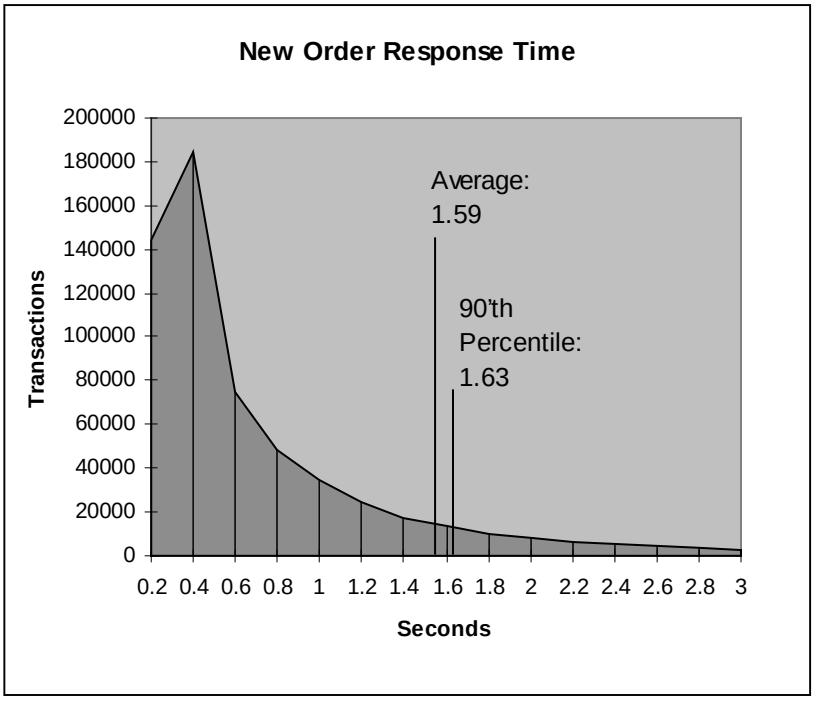


Figure 5.2: Payment Response Time Distribution

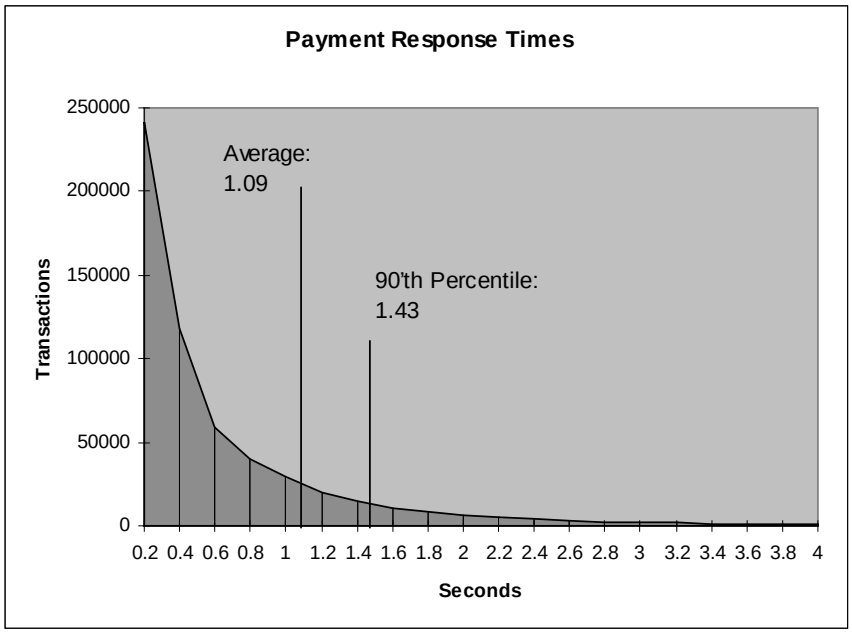
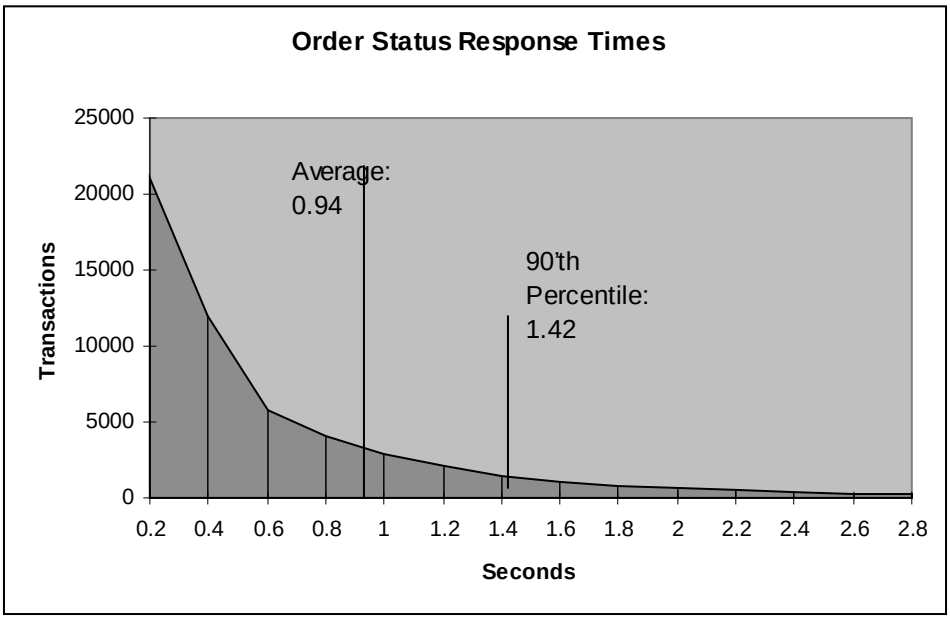


Figure 5.3: Order Status Response Time Distribution



: Delivery Response Time Distribution

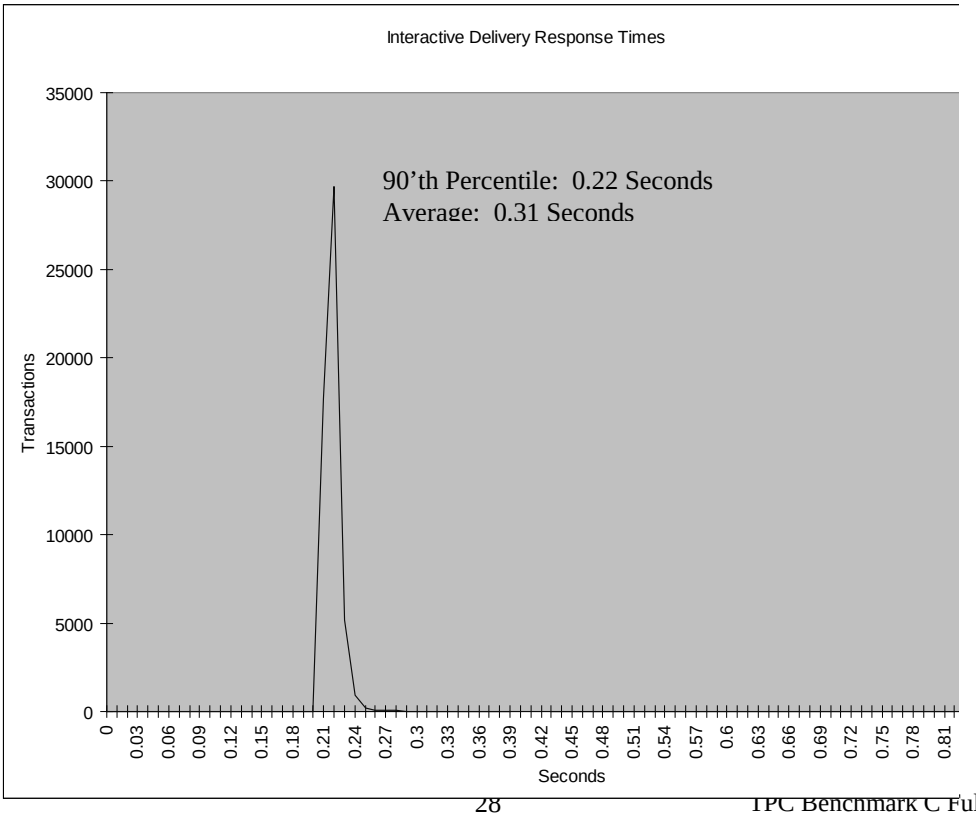


Figure 5.4: Stock Level Response Time Distribution

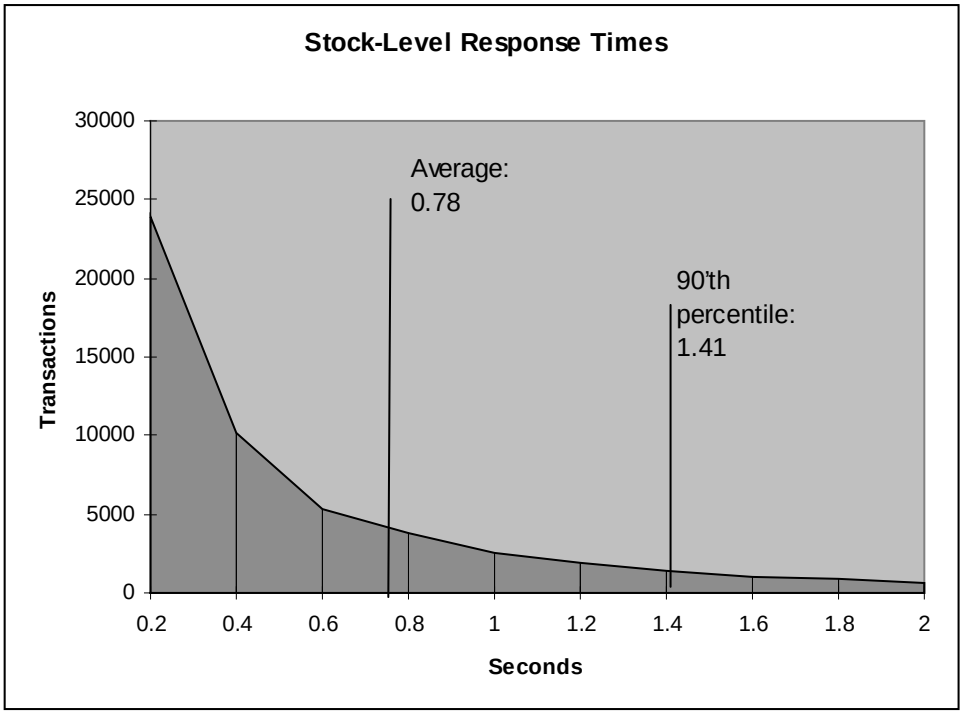


Figure 5.6: New Order Think Time Frequency Distribution

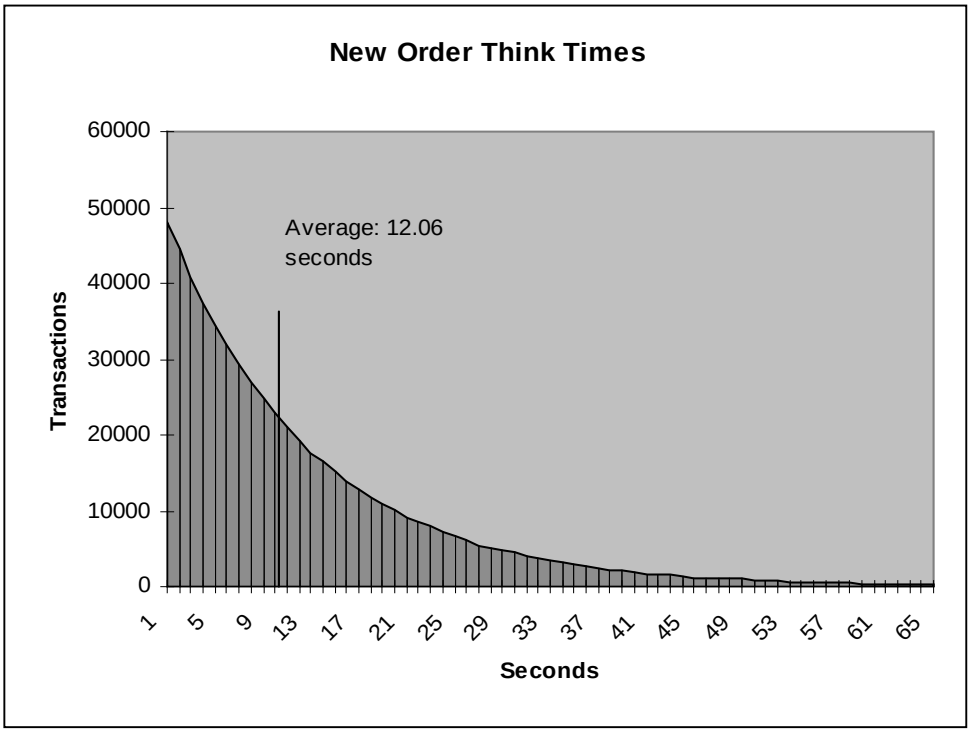


Figure 5.7: Response time versus Throughput

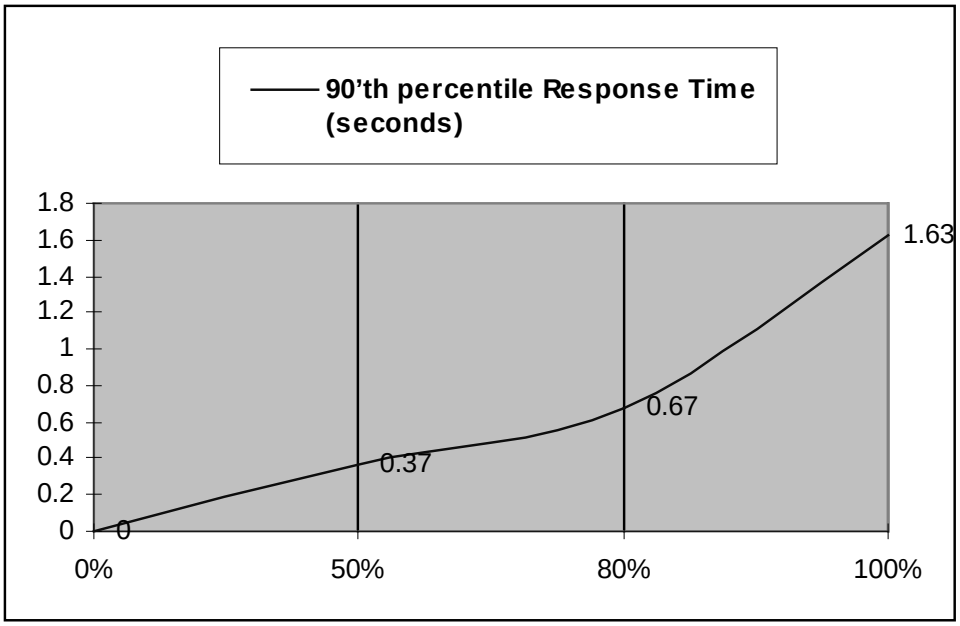
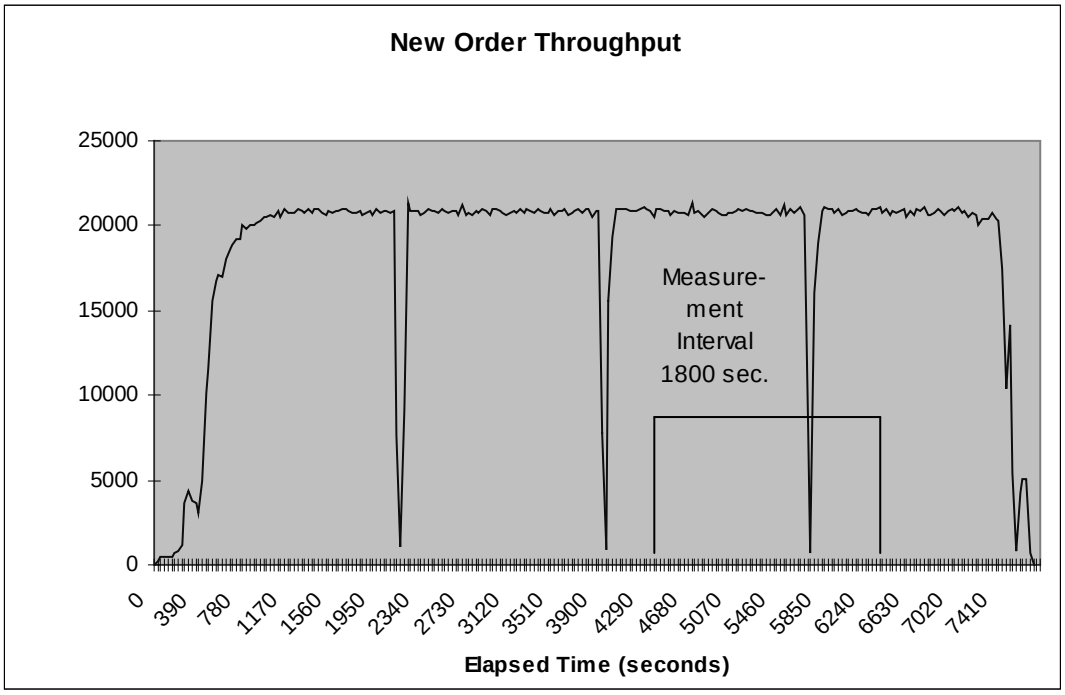


Figure 5.8: New Order Sustained Throughput



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 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 5.1.: Transaction Statistics

Statistics		Value
Transaction Mix	New Order	44.75%
	Payment	43.11%
	Order status	4.05%
	Delivery	4.05%
	Stock level	4.04%
New Order	Home warehouse order lines	99.00%
	Remote warehouse order lines	1.00%
	Rolled back transactions	1.00%
	Average items per order	10.00
Payment	Home warehouse	84.89%
	Remote warehouse	15.11%
	Accessed by last name	60.03%
Order Status	Accessed by last name	60.18%
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.

Two checkpoints were recorded before the measured window opened and another checkpoint was started 1045 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. Eighteen 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 componenets will be available no later than January 21'st, 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:	20,170.63
Price per tpmC	$110.30
Available	January 21'st, 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 200 user license. There were 168 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 Tom Sawyer of Performance Metrics, Inc.

Performance Metrics, Inc.
2229 Benita Dr. Suite 101
Rancho Cordova, CA
(phone) 916/635-2822
(fax) 916/858-0109

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

```
********** 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

********** dummy.c **********
/*
        dummy.c : functions for test.


        Version     1.00        1996/12/26
        Version     1.10        1997/04/26 add
get_alphastr()
        Version     1.20        1998/05/27 for
Tc6/SymfoWARE
        Version     1.21        1998/06/05 delete
'#inlcude "ui.h"'
        Version     1.22        1998/06/13 add
userlog()
        Version     1.23        1998/06/16 add
srand(getpid()) in tpinit() ...
        Version     1.23a       1998/06/18 add
comment for tpinit()



        (C)Fujitsu Limited. 1994, 1995, 1996
*/

#ifdef      SCRTEST

#include            <stdio.h>
#include            <stdlib.h>
#include            <string.h>
#include            <sys/types.h>
#include            <time.h>
#include            <stdarg.h>

#include            "misc.h"
#include            "bench2.h"
#include            "dummy.h"
#include            "Tc.h"

void dummy_delivery( delivery_trans * );
void dummy_stocklevel( stocklvl_trans * );
void dummy_orderstatus( orderstat_trans * );
void dummy_payment( payment_trans * );
void dummy_neworder( neworder_trans * );
char *get_datetimestr( char * );
char *get_datestr( char * );
char *get_alphastr( char *, int );

int         tperrno = 0;


int userlog( char *fmt, ... )
{
        static FILE  *log = NULL;
        char                    fpath[32];
        va_list                 ap;
        int                     rtn;

        if ( log == NULL ) {
                sprintf( fpath,
"/tmp/tcerror.%05d", T_id );
                if( ( log = fopen( fpath, "w" ) )
== NULL ) {
                        return 0;
                }
                setvbuf( log, NULL, _IONBF, 0
);
        }

        va_start( ap, fmt );
        rtn = vfprintf( log, fmt, ap );
        va_end( ap );

        return rtn;
}


int tpcall( char *svc, char *in, long isz, char **out, long
*osz, long mode )
{
        switch ( *( (int *)in ) ) {
        case        TX_NEWORDER:
                    dummy_neworder( (
neworder_trans * )in );
                    break;
        case        TX_PAYMENT:
                    dummy_payment( (
payment_trans * )in );
                    break;
        case        TX_ORDERSTATUS:
                    dummy_orderstatus( (
orderstat_trans * )in );
                    break;
        case        TX_STOCKLEVEL:
                    dummy_stocklevel( (
stocklvl_trans * )in );
                    break;
        }

        *out = in;
        *osz = isz;

        return 0;
}


int tpacall( char *svc, char *in, long isz, long mode )
{
        switch ( *( (int *)in ) ) {
        case        TX_DELIVERY:
```

```
                        dummy_delivery( (
delivery_trans * )in );
                        break;
                }

                return 0;
}


void *tpalloc( char *mode, char *ptr, long size )
{
                return malloc( size );
}


int tpinit( char *ptr )
{
                /* initialize randomize number for dummy
routine */
                srand48( getpid() );        /*
(^^;;;;;; */
                return 0;
}


int tpterm()
{
                return 0;
}


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;
}


char *get_alphastr( char *buf, int len )
{
                int             i;
                int             ch;

                for ( i = 0; i < len-1; i++ ) {
                        buf[i] = rand()%95 + 0x20;
                }
                buf[len-1] = '\0';

                return buf;
}


void dummy_delivery( delivery_trans *bp )
{
                bp->C_R = NOERR;

                return;
}


void dummy_stocklevel( stocklvl_trans *bp )
{
                int             i;

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

                bp->low_stock = i;

                return;
}


void dummy_payment( payment_trans *bp )
{
                bp->C_R = NOERR;

                /* get_datetimestr( bp->h_date ); */
                bp->h_date = time( NULL );
                strcpy( bp->w_street_1, "Baker street" );
                strcpy( bp->w_street_2, "221B" );
                strcpy( bp->w_city, "London" );
                strcpy( bp->w_state, "GB" );
                sprintf( bp->w_zip, "%04d11111",
rand()%10000 );

                strcpy( bp->d_street_1, "Minato-ku" );
                strcpy( bp->d_street_2, "Azabu 10" );
                strcpy( bp->d_city, "Tokyo" );
                strcpy( bp->d_state, "JP" );
                sprintf( bp->d_zip, "%04d11111",
rand()%10000 );

                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" );
                bp->c_discount = 0.20;
                strcpy( bp->c_city, "London" );
                strcpy( bp->c_state, "GB" );
                sprintf( bp->c_zip, "%04d11111",
rand()%10000 );
                sprintf( bp->c_phone,
"%04d%04d%04d%04d",
                        rand()%10000, rand()%10000,
rand()%10000, rand()%10000 );
                bp->c_balance = ( (
rand()*rand()%19999999 )-9999999 ) / 100.0;
                bp->c_credit_lim = 77777;
                /* get_datestr( bp->c_since ); */
                bp->c_since = time( NULL );

                strcpy( bp->c_data,
"Migyamigyamigyamigyamigya"

                "migyamigyamigyamigyamigya" );

                return;
}


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

                bp->C_R = NOERR;

                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 ) / 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 ); */
                bp->o_entry_d = time( NULL );
                bp->o_carrier_id = rand()%100;
                if ( rand()%10 == 0 ) {
                        bp->o_carrier_id = INTNULL;
                }

                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;
                        bp->ol_amount[i] = rand();

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

                        /* get_datetimestr( bp-
>ol_delivery_d[i] ); */
                        bp->ol_delivery_d[i] = time(
NULL );
                }

                return;
}


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

```
        int i;

        bp->C_R = NOERR;
        /* *( bp->status ) = '\0'; */

        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 ); */
        bp->o_entry_d = time( NULL );
        bp->c_discount = ( rand()%101 )/10000.0;
        bp->w_tax = ( rand()%2001 )/10000.0;
        bp->d_tax = ( rand()%2001 )/10000.0;

        bp->total_amount = 0;

        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->newout.i_name[i][0] =
'\0'; */
                get_alphastr( bp->i_name[i],
19 );
                bp->s_quantity[i] = ( rand()%10
)+1;
                bp->brand_generic[i] = (
rand()%26 )+'A';
                bp->i_price[i] = ((
rand()%10000 )+1 )/100.0;
                bp->ol_amount[i]
                        = bp-
>i_price[i]*bp->ol_quantity[i];
                bp->total_amount += bp-
>ol_amount[i];
        }
        bp->o_ol_cnt = i;

        return;
}

#endif

********** dummy.h **********
/*
        dummy.h : functions for test.

        Version 1.00    1997/08/07 for
dummy tuxedo call
        Version 1.01    1998/06/05 add
_DUMMY_H_ (^^;
        Version 1.02    1998/06/13 add
userlog()
*/

#ifndef _DUMMY_H_

#define _DUMMY_H_

#ifdef  SCRTEST

#define TPELIMIT   1
#define TPETIME         2
#define TPGOTSIG 3
#define TPESVCERR       4
#define TPESVCFAIL      5

#define TPSIGRSTRT      1
#define TPNOREPLY       2

/* function prototype */

int userlog( char *, ... );
int tpinit( char * );
void *tpalloc( char *, char *, long );
int tpcall( char *, char *, long, char **, long *, long );
int tpacall( char *, char *, long, long );
int tpterm();

/* global variables */

extern int      tperrno;

#endif  /* SCRTEST */

#endif  /* _DUMMY_H_ */

********** fldtbl **********
#
#               Field Table for TPC-C
#
FML_TERM        2001    long    -
        -
FML_TRAN 2002   long    -       -
FML_DATA 2003   carray  -       -
********** frame.c **********
/*
        frame.c

        Version 1.00    1998/06/05 first
edition
        Version 1.01    1998/06/11 bug
fix for o_ol_cnt
        Version 1.02    1998/06/12 bug
fix for Neworder rollback
        Version 1.03    1998/06/12 add
outputting error messages
        Version 1.03a   1998/06/13
        change debug level
        Version 1.04    1998/06/16 fix
Neworder[7] for DUR mode
        Version 1.05    1998/06/19
        change sqlerror() interface
        Version 1.07    1998/06/26 add
range check (rangeerror())

        into query_delivery_frame(),

        change check_delivery_frame() interface

        for detecting out of range
        Version 1.08    1998/06/27 add
range check into query_stocklevel

        _frame(), check_stocklevel_frame()
*/

#include    <stdio.h>
#include    <stdlib.h>
#include    <string.h>
#include    <sys/time.h>

#include    "misc.h"
#include    "bench2.h"
#include    "Tc.h"

extern const textfield_t   Neworder_text[];
extern datafield_t         Neworder_data[];
extern const textfield_t   Payment_text[];
extern datafield_t         Payment_data[];
extern const textfield_t   Orderstatus_text[];
extern datafield_t
        Orderstatus_data[];
extern const textfield_t   Delivery_text[];
extern datafield_t         Delivery_data[];
extern const textfield_t   Stocklevel_text[];
extern datafield_t         Stocklevel_data[];
extern const int           Start_field[];


#ifdef      SOLARIS2
# define    GETTIMEOFDAY( x ) \
                { \
                        void        *_ptr
= NULL; \
                        gettimeofday( (x),
_ptr ); \
                }
#else       /* !SOLARIS2 */
# define    GETTIMEOFDAY( x )   gettimeofday( x )
#endif      /* !SOLARIS2 */



int assign_neworder_frame( neworder_trans *nt,
datafield_t *nd )
{
        int         idx = 0;
        int         i;

        nd[idx++].data = ( char * )&nt->w_id;
        nd[idx++].data = ( char * )&nt->d_id;
        nd[idx++].data = NULL;
                /* o_entry_d */
        nd[idx++].data = ( char * )&nt->c_id;
        nd[idx++].data = nt->c_last;
        nd[idx++].data = nt->c_credit;
        nd[idx++].data = ( char * )&nt->c_discount;
#ifdef      DUR
        nd[idx++].data = NULL;
                /* o_id */
#else
        nd[idx++].data = ( char * )&nt->o_id;
#endif
        nd[idx++].data = ( char * )&nt->o_ol_cnt;
        nd[idx++].data = ( char * )&nt->w_tax;
        nd[idx++].data = ( char * )&nt->d_tax;

        for ( i = 0; i < 15; i++ ) {
                nd[idx++].data = ( char * )&nt-
>ol_supply_w_id[i];
                nd[idx++].data = ( char * )&nt-
>ol_i_id[i];
                nd[idx++].data = nt->i_name[i];
```

```
                        nd[idx++].data = ( char * )&nt->ol_quantity[i];
                        nd[idx++].data = ( char * )&nt->s_quantity[i];
                        nd[idx++].data = NULL;
                        /* brand_generic[i] */
                        nd[idx++].data = ( char * )&nt->i_price[i];
                        nd[idx++].data = ( char * )&nt->ol_amount[i];
        }

        nd[idx++].data = "\0";
        nd[idx++].data = ( char * )&nt->total_amount;

        return idx;
}


void query_neworder_frame( neworder_trans *nt )
{
        int             i;

        assign_neworder_frame( nt, Neworder_data );

        memset( nt, 0, sizeof( neworder_trans ) );
        nt->C_R = 0;
        nt->tx_type = TX_NEWORDER;
        nt->w_id = W_id;
        Neworder_data[2].data = "";
                /* o_entry_d */
        for ( i = 0; i < 15; i++ ) {
        /* brand_generic */
                Neworder_data[16+i*8].data = "";
        }
#ifdef  DUR
        Neworder_data[7].len = 0;
        Neworder_data[7].data = "";
#endif

        display_frame( FR_FULL, Neworder_text, Neworder_data );
        change_status( "New Order" );
                /* for compatibility */
        refresh();

        query_frame( Start_field[TX_NEWORDER], Neworder_data );
        while ( ( i = check_neworder_frame() ) != CHECKOK ) {
                errorstatus( "Neworder" );
                query_frame( i, Neworder_data );
        }

        for ( i = 0; i < 15; i++ ) {
                if ( nt->ol_supply_w_id[i] == 0 ) {
                        break;
                }
        }
        nt->o_ol_cnt = i;
}


void display_neworder_frame( neworder_trans *nt )
{
        int             i;
        char            o_entry_d[21];
        char            brand_generic[15][2];
#ifdef  DUR
        char            o_id[16] = "ber: ";
#endif

        assign_neworder_frame( nt, Neworder_data );
        if ( nt->C_R == 1 ) {
                for ( i = 0; i < nt->o_ol_cnt; i++ ) {
                        brand_generic[i][1] = '\0';

                Neworder_data[16+i*8].data = brand_generic[i];
                        *( Neworder_data[16+i*8].data ) = nt->brand_generic[i];
                }
                for ( ; i < 15; i++ ) {

                Neworder_data[16+i*8].data = "";
                }
                Neworder_data[2].data = o_entry_d;
                /* o_entry_d */
                convert_datetime( Neworder_data[2].data, nt->o_entry_d );
        } else if ( nt->C_R == 2 ) {
                for ( i = 0; i < 15; i++ ) {
        /* brand_generic */

                Neworder_data[16+i*8].data = "";
                }
                Neworder_data[2].data = "";
                        /* o_entry_d */
                Neworder_data[131].data = "Item number is not valid";
        } else {
                sqlerror( TX_NEWORDER, ( char * )nt );
                return;
        }
#ifdef  DUR
        Neworder_data[7].len = 13;
        Neworder_data[7].data = o_id;
        sprintf( o_id + 5, "%8d", nt->o_id );
#endif

        display_frame( FR_DATA, Neworder_text, Neworder_data );
}


int assign_payment_frame( payment_trans *pt, datafield_t *pd )
{
        int             idx = 0;

        pd[idx++].data = NULL;
        pd[idx++].data = ( char * )&pt->w_id;
        pd[idx++].data = ( char * )&pt->d_id;
        pd[idx++].data = pt->w_street_1;
        pd[idx++].data = pt->d_street_1;
        pd[idx++].data = pt->w_street_2;
        pd[idx++].data = pt->d_street_2;
        pd[idx++].data = pt->w_city;
        pd[idx++].data = pt->w_state;
        pd[idx++].data = NULL;
        pd[idx++].data = pt->d_city;
        pd[idx++].data = pt->d_state;
        pd[idx++].data = NULL;

        pd[idx++].data = ( char * )&pt->c_id;
        pd[idx++].data = ( char * )&pt->c_w_id;
        pd[idx++].data = ( char * )&pt->c_d_id;
        pd[idx++].data = pt->c_first;
        pd[idx++].data = pt->c_middle;
        pd[idx++].data = pt->c_last;
        pd[idx++].data = NULL;
        pd[idx++].data = pt->c_street_1;
        pd[idx++].data = pt->c_credit;
        pd[idx++].data = pt->c_street_2;
        pd[idx++].data = ( char * )&pt->c_discount;
        pd[idx++].data = pt->c_city;
        pd[idx++].data = pt->c_state;
        pd[idx++].data = NULL;
        pd[idx++].data = NULL;
        pd[idx++].data = ( char * )&pt->h_amount;
        pd[idx++].data = ( char * )&pt->c_balance;
        pd[idx++].data = ( char * )&pt->c_credit_lim;

        pd[idx++].data = NULL;
        pd[idx++].data = NULL;
        pd[idx++].data = NULL;
        pd[idx++].data = NULL;

        return idx;
}


void query_payment_frame( payment_trans *pt )
{
        int             i;

        assign_payment_frame( pt, Payment_data );

        memset( pt, 0, sizeof( payment_trans ) );
        pt->C_R = 0;
        pt->tx_type = TX_PAYMENT;
        pt->w_id = W_id;
        Payment_data[0].data = "";
                /* h_amount */
        Payment_data[9].data = "";
                /* w_zip */
        Payment_data[12].data = "";
                /* d_zip */
        Payment_data[26].data = "";
                /* c_zip */
        Payment_data[27].data = "";
                /* c_phone */
        Payment_data[19].data = "";
                /* c_since */
        Payment_data[31].data = "";
                /* c_data1 */
        Payment_data[32].data = "";
                /* c_data2 */
        Payment_data[33].data = "";
                /* c_data3 */
        Payment_data[34].data = "";
                /* c_data4 */

        display_frame( FR_FULL, Payment_text, Payment_data );
```

```
	change_status( "Payment" );
		/* for compatibility */
	refresh();

	query_frame( Start_field[TX_PAYMENT], Payment_data );
	while ( ( i = check_payment_frame() ) != CHECKOK ) {
		errorstatus( "Payment" );
		query_frame( i, Payment_data );
	}
}


void display_payment_frame( payment_trans *pt )
{
	int		i;
	char		h_date[20];
	char		w_zip[11];
	char		d_zip[11];
	char		c_zip[11];
	char		c_phone[20];
	char		c_since[11];
	char		c_data1[51];
	char		c_data2[51];
	char		c_data3[51];
	char		c_data4[51];

	if ( pt->C_R != NOERR ) {
		sqlerror( TX_PAYMENT, ( char * )pt );
		return;
	}

	assign_payment_frame( pt, Payment_data );

	Payment_data[9].data = w_zip;
	Payment_data[26].data = c_zip;
	Payment_data[12].data = d_zip;
	for ( i = 0; i < 5; i++ ) {
		*( Payment_data[9].data + i ) = *( pt->w_zip + i );
		*( Payment_data[12].data + i ) = *( pt->d_zip + i );
		*( Payment_data[26].data + i ) = *( pt->c_zip + i );
	}
	*( Payment_data[9].data + 5 ) = '-';
	*( Payment_data[12].data + 5 ) = '-';
	*( Payment_data[26].data + 5 ) = '-';
	for ( i = 5; i < 9; i++ ) {
		*( Payment_data[9].data + i + 1 ) = *( pt->w_zip + i );
		*( Payment_data[12].data + i + 1 ) = *( pt->d_zip + i );
		*( Payment_data[26].data + i + 1 ) = *( pt->c_zip + i );
	}
	*( Payment_data[9].data + 10 ) = '\0';
	*( Payment_data[12].data + 10 ) = '\0';
	*( Payment_data[26].data + 10 ) = '\0';

	Payment_data[27].data = c_phone;
	for ( i = 0; i < 6; i++ ) {
		*( Payment_data[27].data + i ) = *( pt->c_phone + i );
	}
	*( Payment_data[27].data + 6 ) = '-';
	for ( i = 6; i < 9; i++ ) {
		*( Payment_data[27].data + i + 1 ) = *( pt->c_phone + i );
	}
	*( Payment_data[27].data + 10 ) = '-';
	for ( i = 9; i < 12; i++ ) {
		*( Payment_data[27].data + i + 2 ) = *( pt->c_phone + i );
	}
	*( Payment_data[27].data + 14 ) = '-';
	for ( i = 12; i < 16; i++ ) {
		*( Payment_data[27].data + i + 3 ) = *( pt->c_phone + i );
	}
	*( Payment_data[27].data + 19 ) = '\0';

	Payment_data[0].data = h_date;
	convert_datetime( Payment_data[0].data, pt->h_date );
	Payment_data[19].data = c_since;
	convert_date( Payment_data[19].data, pt->c_since );

	Payment_data[31].data = c_data1;
	*( Payment_data[31].data ) = '\0';
	*( Payment_data[31].data + 50 ) = '\0';
	Payment_data[32].data = c_data2;
	*( Payment_data[32].data ) = '\0';
	*( Payment_data[32].data + 50 ) = '\0';
	Payment_data[33].data = c_data3;
	*( Payment_data[33].data ) = '\0';
	*( Payment_data[33].data + 50 ) = '\0';
	Payment_data[34].data = c_data4;
	*( Payment_data[34].data ) = '\0';
	*( Payment_data[34].data + 50 ) = '\0';

	i = strlen( pt->c_data );
	if ( i > 0 ) {
		strncpy( Payment_data[31].data, pt->c_data, 50 );
		if ( i > 50 ) {
			strncpy( Payment_data[32].data, &pt->c_data[50], 50 );
			if ( i > 100 ) {
	strncpy( Payment_data[33].data,
	&pt->c_data[100], 50 );
				if ( i > 150 ) {
	strncpy( Payment_data[34].data,
	&pt->c_data[150], 50 );
				}
			}
		}
	}

	display_frame( FR_DATA, Payment_text, Payment_data );
}


int assign_orderstatus_frame( orderstat_trans *ot, datafield_t *od )
{
	int		idx = 0;
	int		i;

	od[idx++].data = ( char * )&ot->w_id;
	od[idx++].data = ( char * )&ot->d_id;
	od[idx++].data = ( char * )&ot->c_id;
	od[idx++].data = ot->c_first;
	od[idx++].data = ot->c_middle;
	od[idx++].data = ot->c_last;
	od[idx++].data = ( char * )&ot->c_balance;

	od[idx++].data = ( char * )&ot->o_id;
	od[idx++].data = NULL;
		/* o_entry_d */
	od[idx++].data = ( char * )&ot->o_carrier_id;

	for ( i = 0; i < 15; i++ ) {
		od[idx++].data = ( char * )&ot->ol_supply_w_id[i];
		od[idx++].data = ( char * )&ot->ol_i_id[i];
		od[idx++].data = ( char * )&ot->ol_quantity[i];
		od[idx++].data = ( char * )&ot->ol_amount[i];
		od[idx++].data = NULL;
		/* ol_delivery_d */
	}

	return idx;
}


void query_orderstatus_frame( orderstat_trans *ot )
{
	int		i;

	assign_orderstatus_frame( ot, Orderstatus_data );

	memset( ot, 0, sizeof( orderstat_trans ) );
	ot->C_R = 0;
	ot->tx_type = TX_ORDERSTATUS;
	ot->w_id = W_id;
	Orderstatus_data[8].data = "";
		/* o_entry_d */
	for ( i = 0; i < 15; i++ ) {
		Orderstatus_data[14+i*5].data = "";	/* ol_delivery_d */
	}

	display_frame( FR_FULL, Orderstatus_text, Orderstatus_data );
	change_status( "OrderStatus" );
		/* for compatibility */
	refresh();

	query_frame( Start_field[TX_ORDERSTATUS], Orderstatus_data );
	while ( ( i = check_orderstatus_frame() ) != CHECKOK ) {
		errorstatus( "OrderStatus" );
		query_frame( i, Orderstatus_data );
	}
}


void display_orderstatus_frame( orderstat_trans *ot )
{
```

```
	int		i;
	int		zero = 0;
	char		o_entry_d[20];
	char		ol_delivery_d[15][11];

	if ( ot->C_R != NOERR ) {
		sqlerror( TX_ORDERSTATUS,
( char * )ot );
		return;
	}

	assign_orderstatus_frame( ot,
Orderstatus_data );
	Orderstatus_data[8].data = o_entry_d;
	convert_datetime(
Orderstatus_data[8].data, ot->o_entry_d );
	if ( ot->o_carrier_id == INTNULL ) {
		debug4( ( stderr, "\t INTNULL
found\n" ) );
		Orderstatus_data[9].data = (
char * )&zero;
	}
	for ( i = 0; i < ot->o_ol_cnt; i++ ) {
		Orderstatus_data[14+i*5].data
= ol_delivery_d[i];
		convert_date(
Orderstatus_data[14+i*5].data,
				ot-
>ol_delivery_d[i] );
	}
	for ( ; i < 15; i++ ) {
		Orderstatus_data[14+i*5].data
= "";
	}

	display_frame( FR_DATA,
Orderstatus_text, Orderstatus_data );
}


int assign_delivery_frame( delivery_trans *dt,
datafield_t *dd )
{
	int		idx = 0;

	dd[idx++].data = ( char * )&dt->w_id;
	dd[idx++].data = ( char * )&dt->o_carrier_id;
	dd[idx++].data = "";

	return idx;
}


void query_delivery_frame( delivery_trans *dt )
{
	int			i;
	struct timeval		timeque;

	assign_delivery_frame( dt, Delivery_data );

	memset( dt, 0, sizeof( delivery_trans ) );
	dt->C_R = 0;
	dt->tx_type = TX_DELIVERY;
	dt->w_id = W_id;
	*( Delivery_data[2].data ) = '\0';
		/* status */

	display_frame( FR_FULL, Delivery_text,
Delivery_data );
	change_status( "Delivery" );
		/* for compatibility */
	refresh();

	query_frame( Start_field[TX_DELIVERY],
Delivery_data );
	while ( ( i = check_delivery_frame() ) !=
CHECKOK ) {
		if ( i < 0 ) {
			rangeerror(
"Delivery" );
			i = -i;
		} else {
			errorstatus(
"Delivery" );
		}
		query_frame( i, Delivery_data
);
	}

	GETTIMEOFDAY( &timeque );
	dt->startsec  = timeque.tv_sec;
	dt->startusec = timeque.tv_usec;
}


void display_delivery_frame( delivery_trans *dt )
{
	assign_delivery_frame( dt, Delivery_data );
	Delivery_data[2].data = "Delivery has been
queued";
	display_frame( FR_DATA, Delivery_text,
Delivery_data );
}


int assign_stocklevel_frame( stocklvl_trans *st,
datafield_t *sd )
{
	int		idx = 0;

	sd[idx++].data = ( char * )&st->w_id;
	sd[idx++].data = ( char * )&st->d_id;
	sd[idx++].data = ( char * )&st->threshold;
	sd[idx++].data = ( char * )&st->low_stock;

	return idx;
}


void query_stocklevel_frame( stocklvl_trans *st )
{
	int		i;

	assign_stocklevel_frame( st,
Stocklevel_data );

	memset( st, 0, sizeof( stocklvl_trans ) );
	st->C_R = 0;
	st->tx_type = TX_STOCKLEVEL;
	st->w_id = W_id;
	st->d_id = D_id;

	display_frame( FR_FULL, Stocklevel_text,
Stocklevel_data );
	change_status( "Stock Level" );
		/* for compatibility */
	refresh();

	query_frame(
Start_field[TX_STOCKLEVEL], Stocklevel_data );
	while ( ( i = check_stocklevel_frame() ) !=
CHECKOK ) {
		if ( i < 0 ) {
			rangeerror( "Stock
Level" );
			i = -i;
		} else {
			errorstatus( "Stock
Level" );
		}
		query_frame( i,
Stocklevel_data );
	}
}


void display_stocklevel_frame( stocklvl_trans *st )
{
	if ( st->C_R != NOERR ) {
#ifndef	AVOID_SVC_BUG
		sqlerror( TX_STOCKLEVEL, (
char * )st );
#endif
		return;
	}

	assign_stocklevel_frame( st,
Stocklevel_data );
	display_frame( FR_DATA, Stocklevel_text,
Stocklevel_data );
}


int read_value( datafield_t *df )
{
	if ( df->flag & FL_INT ) {
		return *( ( int * )df->data );
	} else if ( df->flag & FL_SHORT ) {
		return *( ( short * )df->data );
	} else if ( df->flag & FL_STR ) {
		return  *df->data;
		/* first char */
	} else if ( df->flag & FL_DBL ) {
		return *( ( double * )df->data );
	}

	return 0;
}


int check_neworder_frame()
{
	int		sw, i, q;
	int		j;

	if ( !read_value( &Neworder_data[1] ) ) {
		debug( ( stderr, "\t %d: no field
data in neworder\n", 1 ) );
		return 1;
	}
	if ( !read_value( &Neworder_data[3] ) ) {
		debug( ( stderr, "\t %d: no field
data in neworder\n", 3 ) );
		return 3;
	}

	for ( j = 0; j < 15; j++ ) {
```

```
                        sw = read_value(
&Neworder_data[11+j*8] );
                        i  = read_value(
&Neworder_data[12+j*8] );
                        q  = read_value(
&Neworder_data[14+j*8] );
                        debug4( ( stderr, "\t %2d
s_w_id: %d, i_id: %d, quantity: %d\n",
                                j, sw, i, q ) );

                        if ( ( !sw ) && ( !i ) && ( !q ) ) {
                                break;
                        }

                        if ( !sw ) {
                                debug( ( stderr, "\t
%d: no field data in neworder\n",
                                                11+j*8
) );
                                return 11+j*8;
                        }
                        if ( !q ) {
                                if ( !i ) {

                debug( ( stderr,

                "\t %d: no field data in neworder\n",

                12+j*8 ) );
                                                return
12+j*8;
                                }
                                debug( ( stderr, "\t
%d: no field data in neworder\n",
                                                14+j*8
) );
                                return 14+j*8;
                        }
                }

                if ( j == 0 ) {
                /* no orderline */
                        debug( ( stderr, "\t %d: no field
data in neworder\n", 11 ) );
                        return 11;
                }

                return CHECKOK;
}


int check_payment_frame()
{
                int             i;

                if ( ( i = 2, !read_value( &Payment_data[i] )
)
                        || ( i = 14, !read_value(
&Payment_data[i] ) )
                        || ( i = 15, !read_value(
&Payment_data[i] ) )
                        || ( i = 28, !read_value(
&Payment_data[i] ) ) )
                {
                        debug( ( stderr, "\t %d: no field
data in payment\n", i ) );
                        return i;
                }
                if ( !read_value( &Payment_data[13] )
                        && !read_value(
&Payment_data[18] ) )
                {
                        debug( ( stderr, "\t %d: no field
data in payment\n", 13 ) );
                        return 13;
                }

                return CHECKOK;
}


int check_orderstatus_frame()
{
                if ( !read_value( &Orderstatus_data[1] ) ) {
                        debug( ( stderr, "\t %d: no field
data in orderstatus\n", 1 ) );
                        return 1;
                }
                if ( !read_value( &Orderstatus_data[2] )
                        && !read_value(
&Orderstatus_data[5] ) )
                {
                        debug( ( stderr, "\t %d: no field
data in orderstatus\n", 2 ) );
                        return 2;
                }

                return CHECKOK;
}


int check_delivery_frame()
{
                int             id;

                if ( !( id = read_value( &Delivery_data[1] ) )
) {
                        debug( ( stderr, "\t %d: no field
data in delivery%d\n", 1 ) );
                        return 1;
                }
                if ( id > 10 ) {
                        debug( ( stderr, "\t %d: out of
range in delivery\n", 1 ) );
                        return -1;
                }

                return CHECKOK;
}


int check_stocklevel_frame()
{
                int             th;

                if ( !( th = read_value( &Stocklevel_data[2] )
) ) {
                        debug( ( stderr, "\t %d: no field
data in stocklevel\n", 2 ) );
                        return 2;
                }
                if ( th < 10 || th > 20 ) {
                        debug( ( stderr, "\t %d: out of
range in stocklevel\n", 2 ) );
                        return -2;
                }
                return CHECKOK;
}

********** layout.c **********
/*
                layout.c

                Version         0.10            1997/07/31
                (frame.c) first test version
                Version         0.20            1997/07/31  detele
len, ptr data...
                Version         0.21            1997/08/07
                variable names changed.
                ---------------------------
                Version         1.00            1998/05/27
                (layout.c) (for Tc6)
                Version         1.01            1998/06/04  const
char *dollar = "$"; / fix term.
                Version         1.02            1998/06/05  length
of cust-balance fixed
*/

#include      <stdio.h>
#include      <stdlib.h>
#include      <string.h>
#include      "ui.h"


#define       FL_BININT
                (FL_INPUT|FL_BLANK|FL_INT)
#define       FL_BINSHORT
                (FL_INPUT|FL_BLANK|FL_SHORT)
#define       FL_ININT    (FL_INPUT|FL_INT)
#define       FL_INSHORT
                (FL_INPUT|FL_SHORT)
#define       FL_INSTR    (FL_INPUT|FL_STR)

#define       FL_BOUTINT
                (FL_DATA|FL_BLANK|FL_INT)
#define       FL_BOUTSHORT
                (FL_DATA|FL_BLANK|FL_SHORT)
#define       FL_OUTINT
                (FL_DATA|FL_INT)
#define       FL_OUTSHORT
                (FL_DATA|FL_SHORT)
#define       FL_OUTSTR
                (FL_DATA|FL_STR)
#define       FL_OUTCHAR
                (FL_DATA|FL_STR)

const char    dollar[2] = "$";

const textfield_t             Delivery_text[] = {
                { 0, 37, "Delivery" },
                { 1,  0, "Warehouse:" },
                { 3,  0, "Carrier Number:" },
                { 5,  0, "Execution Status:" },
                { -1, -1, NULL },
};

datafield_t   Delivery_data[] = {
                { 1, 11,  4, 0, FL_BOUTSHORT, NULL },
                                /* W_ID */
                { 3, 16,  2, 0, FL_BINSHORT, NULL },
                                /* O_CARRIER_ID */
                { 5, 18, 24, 0, FL_OUTSTR, NULL },
                /* Execution Status */
                { -1, -1,  0, 0, NULL, NULL },
};
```

```
const textfield_t           Stocklevel_text[] = {
            { 0, 34, "Stock-Level" },
            { 1,  0, "Warehouse:" },
            { 1, 18, "District:" },
            { 3,  0, "Stock Level Threshold:" },
            { 5,  0, "low stock:" },
            { -1, -1, NULL },
};

datafield_t Stocklevel_data[] = {
            { 1, 11,  4, 0, FL_BOUTSHORT, NULL },
                        /* W_ID */
            { 1, 28,  2, 0, FL_BOUTSHORT, NULL },
                        /* D_ID */
            { 3, 23,  2, 0, FL_ININT, NULL },
            /* Threshold */
            { 5, 11,  3, 0, FL_OUTINT|FL_BLANK,
NULL },     /* low_stock */
            { -1, -1,  0, 0, 0, NULL },
};

const textfield_t           Payment_text[] = {
            { 0, 37, "Payment" },
            { 1,  0, "Date:" },
            { 3,  0, "Warehouse:" },
            { 3, 41, "District:" },

            { 8,  0, "Customer:" },
            { 8, 16, "Cust-Warehouse:" },
            { 8, 38, "Cust-District:" },
            { 9,  0, "Name:" },
            { 9, 49, "Since:" },
            { 10, 49, "Credit:" },
            { 11, 49, "%Disc:" },
            { 12, 49, "Phone:" },
            { 14, 0, "Amount Paid:" },
            { 14, 22, dollar },
            { 14, 36, "New Cust-Balance: $" },
            { 15, 0, "Credit Limit:   $" },

            { 17, 0, "Cust-Data:" },
            { -1, -1, NULL },
};

datafield_t Payment_data[] = {
            { 1,  6, 19, 0, FL_OUTSTR, NULL },
            /* H_DATE */
            { 3, 11,  4, 0, FL_BOUTSHORT, NULL },
                        /* W_ID */
            { 3, 51,  2, 0, FL_BINSHORT, NULL },
                        /* D_ID */
            { 4,  0, 20, 0, FL_OUTSTR, NULL },
            /* W_STREET_1 */
            { 4, 41, 20, 0, FL_OUTSTR, NULL },
            /* D_STREET_1 */
            { 5,  0, 20, 0, FL_OUTSTR, NULL },
            /* W_STREET_2 */
            { 5, 41, 20, 0, FL_OUTSTR, NULL },
            /* D_STREET_2 */
            { 6,  0, 20, 0, FL_OUTSTR, NULL },
            /* W_CITY */
            { 6, 21,  2, 0, FL_OUTSTR, NULL },
            /* W_STATE */
            { 6, 24, 10, 0, FL_OUTSTR, NULL },
            /* W_ZIP */
            { 6, 41, 20, 0, FL_OUTSTR, NULL },
            /* D_CITY */
            { 6, 62,  2, 0, FL_OUTSTR, NULL },
            /* D_STATE */
            { 6, 65, 10, 0, FL_OUTSTR, NULL },
            /* D_ZIP */

            { 8, 10,  4, 0, FL_BININT|FL_DATA, NULL
},          /* C_ID */
            { 8, 32,  4, 0, FL_BINSHORT, NULL },
                        /* C_W_ID */
            { 8, 53,  2, 0, FL_BINSHORT, NULL },
                        /* C_D_ID */
            { 9,  8, 16, 0, FL_OUTSTR, NULL },
            /* C_FIRST */
            { 9, 25,  2, 0, FL_OUTSTR, NULL },
            /* C_MIDDLE */
            { 9, 28, 16, 0, FL_INSTR|FL_DATA, NULL
},          /* C_LAST */
            { 9, 57, 10, 0, FL_OUTSTR, NULL },
            /* C_SINCE */
            {10,  8, 20, 0, FL_OUTSTR, NULL },
            /* C_STREET_1 */
            {10, 57,  2, 0, FL_OUTSTR, NULL },
            /* C_CREDIT */
            {11,  8, 20, 0, FL_OUTSTR, NULL },
            /* C_STREET_2 */
            {11, 57,  5, 2, FL_OUTINT, NULL },
            /* C_DISCOUNT */
            {12,  8, 20, 0, FL_OUTSTR, NULL },
            /* C_CITY */
            {12, 29,  2, 0, FL_OUTSTR, NULL },
            /* C_STATE */
            {12, 32, 10, 0, FL_OUTSTR, NULL },
                        /* C_ZIP */
            {12, 57, 19, 0, FL_OUTSTR, NULL },
                        /* C_PHONE */
            {14, 23,  7, 2, FL_ININT, NULL },
            /* H_AMOUNT */
            {14, 55, 14, 2,
FL_DATA|FL_SIGN|FL_DBL, NULL },/* C_BALANCE */
            {15, 17, 13, 2, FL_DATA|FL_DBL, NULL },
            /* C_CREDIT_LIM */

            {17, 11, 50, 0, FL_OUTSTR, NULL },
                        /* C_DATA */
            {18, 11, 50, 0, FL_OUTSTR, NULL },
                        /* C_DATA */
            {19, 11, 50, 0, FL_OUTSTR, NULL },
                        /* C_DATA */
            {20, 11, 50, 0, FL_OUTSTR, NULL },
                        /* C_DATA */
            {-1, -1,  0, 0, 0, NULL },
};

const textfield_t Orderstatus_text[] = {
            { 0, 34, "Order-Status" },
            { 1,  0, "Warehouse:" },
            { 1, 18, "District:" },
            { 2,  0, "Customer:" },
            { 2, 17, "Name:" },
            { 3,  0, "Cust-Balance: $" },
            { 5,  0, "Order-Number:" },
            { 5, 25, "Entry-Date:" },
            { 5, 59, "Carrier-Number:" },

            { 6,  0, "Supply-W" },
            { 6, 13, "Item-Id" },
            { 6, 24, "Qty" },
            { 6, 32, "Amount" },
            { 6, 44, "Delivery-Date" },

            { 7, 31, dollar },
            { 8, 31, dollar },
            { 9, 31, dollar },
            { 10, 31, dollar },
            { 11, 31, dollar },
            { 12, 31, dollar },
            { 13, 31, dollar },
            { 14, 31, dollar },
            { 15, 31, dollar },
            { 16, 31, dollar },
            { 17, 31, dollar },
            { 18, 31, dollar },
            { 19, 31, dollar },
            { 20, 31, dollar },
            { 21, 31, dollar },
            { -1, -1, NULL },
};

datafield_t Orderstatus_data[] = {
            { 1, 11,  4, 0, FL_BOUTSHORT, NULL },
                        /* W_ID */
            { 1, 28,  2, 0, FL_BINSHORT, NULL },
                        /* D_ID */
            { 2, 10,  4, 0, FL_BININT|FL_DATA, NULL
},          /* C_ID */
            { 2, 23, 16, 0, FL_OUTSTR, NULL },
            /* C_FIRST */
            { 2, 40,  2, 0, FL_OUTSTR, NULL },
            /* C_MIDDLE */
            { 2, 43, 16, 0, FL_INSTR|FL_DATA, NULL
},          /* C_LAST */
            { 3, 15,  9, 2, FL_DATA|FL_SIGN|FL_DBL,
NULL },/* C_BALANCE */

            { 5, 14,  8, 0, FL_BOUTINT, NULL },
                        /* O_ID */
            { 5, 37, 19, 0, FL_OUTSTR, NULL },
            /* O_ENTRY_D */
            { 5, 75,  2, 0, FL_BOUTSHORT, NULL },
                        /* O_CARRIER_ID */

            { 7,  2,  4, 0, FL_BOUTSHORT, NULL },
                        /* OL_SUPPLY_W_ID_1 */
            { 7, 13,  6, 0, FL_BOUTINT, NULL },
            /* OL_I_ID_1 */
            { 7, 24,  2, 0, FL_OUTSHORT, NULL },
                        /* OL_QUANTITY_1 */
            { 7, 32,  8, 2, FL_OUTINT, NULL },
            /* OL_AMOUNT_1 */
            { 7, 46, 10, 0, FL_OUTSTR, NULL },
            /* OL_DELIVERY_D_1 */

            { 8,  2,  4, 0, FL_BOUTSHORT, NULL },
                        /* OL_SUPPLY_W_ID_2 */
            { 8, 13,  6, 0, FL_BOUTINT, NULL },
            /* OL_I_ID_2 */
            { 8, 24,  2, 0, FL_OUTSHORT, NULL },
                        /* OL_QUANTITY_2 */
            { 8, 32,  8, 2, FL_OUTINT, NULL },
            /* OL_AMOUNT_2 */
            { 8, 46, 10, 0, FL_OUTSTR, NULL },
            /* OL_DELIVERY_D_2 */

            { 9,  2,  4, 0, FL_BOUTSHORT, NULL },
                        /* OL_SUPPLY_W_ID_3 */
            { 9, 13,  6, 0, FL_BOUTINT, NULL },
            /* OL_I_ID_3 */
            { 9, 24,  2, 0, FL_OUTSHORT, NULL },
                        /* OL_QUANTITY_3 */
```

```
{ 9, 32,  8, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_3 */
{ 9, 46, 10, 0, FL_OUTSTR, NULL },
/* OL_DELIVERY_D_3 */

{ 10,  2,  4, 0, FL_BOUTSHORT, NULL },
                /* OL_SUPPLY_W_ID_4 */
{ 10, 13,  6, 0, FL_BOUTINT, NULL },
                /* OL_I_ID_4 */
{ 10, 24,  2, 0, FL_OUTSHORT, NULL },
                /* OL_QUANTITY_4 */
{ 10, 32,  8, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_4 */
{ 10, 46, 10, 0, FL_OUTSTR, NULL },
                /* OL_DELIVERY_D_4 */

{ 11,  2,  4, 0, FL_BOUTSHORT, NULL },
                /* OL_SUPPLY_W_ID_5 */
{ 11, 13,  6, 0, FL_BOUTINT, NULL },
                /* OL_I_ID_5 */
{ 11, 24,  2, 0, FL_OUTSHORT, NULL },
                /* OL_QUANTITY_5 */
{ 11, 32,  8, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_5 */
{ 11, 46, 10, 0, FL_OUTSTR, NULL },
                /* OL_DELIVERY_D_5 */

{ 12,  2,  4, 0, FL_BOUTSHORT, NULL },
                /* OL_SUPPLY_W_ID_6 */
{ 12, 13,  6, 0, FL_BOUTINT, NULL },
                /* OL_I_ID_6 */
{ 12, 24,  2, 0, FL_OUTSHORT, NULL },
                /* OL_QUANTITY_6 */
{ 12, 32,  8, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_6 */
{ 12, 46, 10, 0, FL_OUTSTR, NULL },
                /* OL_DELIVERY_D_6 */

{ 13,  2,  4, 0, FL_BOUTSHORT, NULL },
                /* OL_SUPPLY_W_ID_7 */
{ 13, 13,  6, 0, FL_BOUTINT, NULL },
                /* OL_I_ID_7 */
{ 13, 24,  2, 0, FL_OUTSHORT, NULL },
                /* OL_QUANTITY_7 */
{ 13, 32,  8, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_7 */
{ 13, 46, 10, 0, FL_OUTSTR, NULL },
                /* OL_DELIVERY_D_7 */

{ 14,  2,  4, 0, FL_BOUTSHORT, NULL },
                /* OL_SUPPLY_W_ID_8 */
{ 14, 13,  6, 0, FL_BOUTINT, NULL },
                /* OL_I_ID_8 */
{ 14, 24,  2, 0, FL_OUTSHORT, NULL },
                /* OL_QUANTITY_8 */
{ 14, 32,  8, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_8 */
{ 14, 46, 10, 0, FL_OUTSTR, NULL },
                /* OL_DELIVERY_D_8 */

{ 15,  2,  4, 0, FL_BOUTSHORT, NULL },
                /* OL_SUPPLY_W_ID_9 */
{ 15, 13,  6, 0, FL_BOUTINT, NULL },
                /* OL_I_ID_9 */
{ 15, 24,  2, 0, FL_OUTSHORT, NULL },
                /* OL_QUANTITY_9 */
{ 15, 32,  8, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_9 */
{ 15, 46, 10, 0, FL_OUTSTR, NULL },
                /* OL_DELIVERY_D_9 */

{ 16,  2,  4, 0, FL_BOUTSHORT, NULL },
                /* OL_SUPPLY_W_ID_10 */
{ 16, 13,  6, 0, FL_BOUTINT, NULL },
                /* OL_I_ID_10 */
{ 16, 24,  2, 0, FL_OUTSHORT, NULL },
                /* OL_QUANTITY_10 */
{ 16, 32,  8, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_10 */
{ 16, 46, 10, 0, FL_OUTSTR, NULL },
                /* OL_DELIVERY_D_10 */

{ 17,  2,  4, 0, FL_BOUTSHORT, NULL },
                /* OL_SUPPLY_W_ID_11 */
{ 17, 13,  6, 0, FL_BOUTINT, NULL },
                /* OL_I_ID_11 */
{ 17, 24,  2, 0, FL_OUTSHORT, NULL },
                /* OL_QUANTITY_11 */
{ 17, 32,  8, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_11 */
{ 17, 46, 10, 0, FL_OUTSTR, NULL },
                /* OL_DELIVERY_D_11 */

{ 18,  2,  4, 0, FL_BOUTSHORT, NULL },
                /* OL_SUPPLY_W_ID_12 */
{ 18, 13,  6, 0, FL_BOUTINT, NULL },
                /* OL_I_ID_12 */
{ 18, 24,  2, 0, FL_OUTSHORT, NULL },
                /* OL_QUANTITY_12 */
{ 18, 32,  8, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_12 */
{ 18, 46, 10, 0, FL_OUTSTR, NULL },
                /* OL_DELIVERY_D_12 */

{ 19,  2,  4, 0, FL_BOUTSHORT, NULL },
                /* OL_SUPPLY_W_ID_13 */
{ 19, 13,  6, 0, FL_BOUTINT, NULL },
                /* OL_I_ID_13 */
{ 19, 24,  2, 0, FL_OUTSHORT, NULL },
                /* OL_QUANTITY_13 */
{ 19, 32,  8, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_13 */
{ 19, 46, 10, 0, FL_OUTSTR, NULL },
                /* OL_DELIVERY_D_13 */

{ 20,  2,  4, 0, FL_BOUTSHORT, NULL },
                /* OL_SUPPLY_W_ID_14 */
{ 20, 13,  6, 0, FL_BOUTINT, NULL },
                /* OL_I_ID_14 */
{ 20, 24,  2, 0, FL_OUTSHORT, NULL },
                /* OL_QUANTITY_14 */
{ 20, 32,  8, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_14 */
{ 20, 46, 10, 0, FL_OUTSTR, NULL },
                /* OL_DELIVERY_D_14 */

{ 21,  2,  4, 0, FL_BOUTSHORT, NULL },
                /* OL_SUPPLY_W_ID_15 */
{ 21, 13,  6, 0, FL_BOUTINT, NULL },
                /* OL_I_ID_15 */
{ 21, 24,  2, 0, FL_OUTSHORT, NULL },
                /* OL_QUANTITY_15 */
{ 21, 32,  8, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_15 */
{ 21, 46, 10, 0, FL_OUTSTR, NULL },
                /* OL_DELIVERY_D_15 */
                { -1, -1,  0, 0, 0, NULL },
};

const textfield_t           Neworder_text[] = {
                { 0, 35, "New Order" },
                { 1,  0, "Warehouse:" },
                { 1, 18, "District:" },
                { 1, 54, "Date:" },
                { 2,  0, "Customer:" },
                { 2, 18, "Name:" },
                { 2, 43, "Credit:" },
                { 2, 56, "%Disc:" },
                { 3,  0, "Order Number:" },
                { 3, 24, "Number of Lines:" },
                { 3, 51, "W_tax:" },
                { 3, 66, "D_tax:" },

                { 5,  1, "Supp_W" },
                { 5,  9, "Item_Id" },
                { 5, 18, "Item Name" },
                { 5, 44, "Qty" },
                { 5, 49, "Stock" },
                { 5, 56, "B/G" },
                { 5, 61, "Price" },
                { 5, 70, "Amount" },

                { 6, 61, dollar },
                { 6, 70, dollar },
                { 7, 61, dollar },
                { 7, 70, dollar },
                { 8, 61, dollar },
                { 8, 70, dollar },
                { 9, 61, dollar },
                { 9, 70, dollar },
                { 10, 61, dollar },
                { 10, 70, dollar },
                { 11, 61, dollar },
                { 11, 70, dollar },
                { 12, 61, dollar },
                { 12, 70, dollar },
                { 13, 61, dollar },
                { 13, 70, dollar },
                { 14, 61, dollar },
                { 14, 70, dollar },
                { 15, 61, dollar },
                { 15, 70, dollar },
                { 16, 61, dollar },
                { 16, 70, dollar },
                { 17, 61, dollar },
                { 17, 70, dollar },
                { 18, 61, dollar },
                { 18, 70, dollar },
                { 19, 61, dollar },
                { 19, 70, dollar },
                { 20, 61, dollar },
                { 20, 70, dollar },

                { 21,  0, "Execution Status:" },
                { 21, 61, "Total:  $" },
                { -1, -1, NULL },
};

datafield_t  Neworder_data[] = {
                { 1, 11,  4, 0, FL_BOUTSHORT, NULL },
                        /* W_ID */
                { 1, 28,  2, 0, FL_BINSHORT, NULL },
                        /* D_ID */
                { 1, 60, 19, 0, FL_OUTSTR, NULL },
                /* O_ENTRY_D */
```

```
        { 2, 11,  4, 0, FL_BININT, NULL },
        /* C_ID */
        { 2, 24, 16, 0, FL_OUTSTR, NULL },
        /* C_LAST */
        { 2, 51,  2, 0, FL_OUTSTR, NULL },
        /* C_CREDIT */
        { 2, 63,  5, 2, FL_OUTINT, NULL },
        /* C_DISCOUNT */
#ifdef  DUR
        { 3,  9, 13, 0, FL_OUTSTR, NULL },
        /* O_ID */
#else
        { 3, 14,  8, 0, FL_BOUTINT, NULL },
                /* O_ID */
#endif
        { 3, 41,  2, 0, FL_OUTSHORT, NULL },
                /* O_OL_CNT */
        { 3, 58,  5, 2, FL_OUTINT, NULL },
        /* W_TAX */
        { 3, 73,  5, 2, FL_OUTINT, NULL },
        /* D_TAX */

        { 6,  2,  4, 0, FL_BINSHORT, NULL },
                /* OL_SUPPLY_W_ID_1 */
        { 6,  9,  6, 0, FL_BININT, NULL },
        /* OL_I_ID_1 */
        { 6, 18, 24, 0, FL_OUTSTR, NULL },
                /* I_NAME_1 */
        { 6, 44,  2, 0, FL_INSHORT, NULL },
                /* OL_QUANTITY_1 */
        { 6, 50,  3, 0, FL_OUTINT, NULL },
        /* S_QUANTITY_1 */
        { 6, 57,  1, 0, FL_OUTCHAR, NULL },
                /* BRAND_GENERIC_1 */
        { 6, 62,  6, 2, FL_OUTINT, NULL },
        /* I_PRICE_1 */
        { 6, 71,  7, 2, FL_OUTINT, NULL },
        /* OL_AMOUNT_1 */

        { 7,  2,  4, 0, FL_BINSHORT, NULL },
                /* OL_SUPPLY_W_ID_2 */
        { 7,  9,  6, 0, FL_BININT, NULL },
        /* OL_I_ID_2 */
        { 7, 18, 24, 0, FL_OUTSTR, NULL },
                /* I_NAME_2 */
        { 7, 44,  2, 0, FL_INSHORT, NULL },
                /* OL_QUANTITY_2 */
        { 7, 50,  3, 0, FL_OUTINT, NULL },
        /* S_QUANTITY_2 */
        { 7, 57,  1, 0, FL_OUTCHAR, NULL },
                /* BRAND_GENERIC_2 */
        { 7, 62,  6, 2, FL_OUTINT, NULL },
        /* I_PRICE_2 */
        { 7, 71,  7, 2, FL_OUTINT, NULL },
        /* OL_AMOUNT_2 */

        { 8,  2,  4, 0, FL_BINSHORT, NULL },
                /* OL_SUPPLY_W_ID_3 */
        { 8,  9,  6, 0, FL_BININT, NULL },
        /* OL_I_ID_3 */
        { 8, 18, 24, 0, FL_OUTSTR, NULL },
                /* I_NAME_3 */
        { 8, 44,  2, 0, FL_INSHORT, NULL },
                /* OL_QUANTITY_3 */
        { 8, 50,  3, 0, FL_OUTINT, NULL },
        /* S_QUANTITY_3 */
        { 8, 57,  1, 0, FL_OUTCHAR, NULL },
                /* BRAND_GENERIC_3 */

        { 8, 62,  6, 2, FL_OUTINT, NULL },
        /* I_PRICE_3 */
        { 8, 71,  7, 2, FL_OUTINT, NULL },
        /* OL_AMOUNT_3 */

        { 9,  2,  4, 0, FL_BINSHORT, NULL },
                /* OL_SUPPLY_W_ID_4 */
        { 9,  9,  6, 0, FL_BININT, NULL },
        /* OL_I_ID_4 */
        { 9, 18, 24, 0, FL_OUTSTR, NULL },
                /* I_NAME_4 */
        { 9, 44,  2, 0, FL_INSHORT, NULL },
                /* OL_QUANTITY_4 */
        { 9, 50,  3, 0, FL_OUTINT, NULL },
        /* S_QUANTITY_4 */
        { 9, 57,  1, 0, FL_OUTCHAR, NULL },
                /* BRAND_GENERIC_4 */
        { 9, 62,  6, 2, FL_OUTINT, NULL },
        /* I_PRICE_4 */
        { 9, 71,  7, 2, FL_OUTINT, NULL },
        /* OL_AMOUNT_4 */

        {10,  2,  4, 0, FL_BINSHORT, NULL },
                /* OL_SUPPLY_W_ID_5 */
        {10,  9,  6, 0, FL_BININT, NULL },
        /* OL_I_ID_5 */
        {10, 18, 24, 0, FL_OUTSTR, NULL },
                /* I_NAME_5 */
        {10, 44,  2, 0, FL_INSHORT, NULL },
                /* OL_QUANTITY_5 */
        {10, 50,  3, 0, FL_OUTINT, NULL },
        /* S_QUANTITY_5 */
        {10, 57,  1, 0, FL_OUTCHAR, NULL },
                /* BRAND_GENERIC_5 */
        {10, 62,  6, 2, FL_OUTINT, NULL },
        /* I_PRICE_5 */
        {10, 71,  7, 2, FL_OUTINT, NULL },
        /* OL_AMOUNT_5 */

        {11,  2,  4, 0, FL_BINSHORT, NULL },
                /* OL_SUPPLY_W_ID_6 */
        {11,  9,  6, 0, FL_BININT, NULL },
        /* OL_I_ID_6 */
        {11, 18, 24, 0, FL_OUTSTR, NULL },
                /* I_NAME_6 */
        {11, 44,  2, 0, FL_INSHORT, NULL },
                /* OL_QUANTITY_6 */
        {11, 50,  3, 0, FL_OUTINT, NULL },
        /* S_QUANTITY_6 */
        {11, 57,  1, 0, FL_OUTCHAR, NULL },
                /* BRAND_GENERIC_6 */
        {11, 62,  6, 2, FL_OUTINT, NULL },
        /* I_PRICE_6 */
        {11, 71,  7, 2, FL_OUTINT, NULL },
        /* OL_AMOUNT_6 */

        {12,  2,  4, 0, FL_BINSHORT, NULL },
                /* OL_SUPPLY_W_ID_7 */
        {12,  9,  6, 0, FL_BININT, NULL },
        /* OL_I_ID_7 */
        {12, 18, 24, 0, FL_OUTSTR, NULL },
                /* I_NAME_7 */
        {12, 44,  2, 0, FL_INSHORT, NULL },
                /* OL_QUANTITY_7 */
        {12, 50,  3, 0, FL_OUTINT, NULL },
        /* S_QUANTITY_7 */
        {12, 57,  1, 0, FL_OUTCHAR, NULL },
                /* BRAND_GENERIC_7 */

        {12, 62,  6, 2, FL_OUTINT, NULL },
        /* I_PRICE_7 */
        {12, 71,  7, 2, FL_OUTINT, NULL },
        /* OL_AMOUNT_7 */

        {13,  2,  4, 0, FL_BINSHORT, NULL },
                /* OL_SUPPLY_W_ID_8 */
        {13,  9,  6, 0, FL_BININT, NULL },
        /* OL_I_ID_8 */
        {13, 18, 24, 0, FL_OUTSTR, NULL },
                /* I_NAME_8 */
        {13, 44,  2, 0, FL_INSHORT, NULL },
                /* OL_QUANTITY_8 */
        {13, 50,  3, 0, FL_OUTINT, NULL },
        /* S_QUANTITY_8 */
        {13, 57,  1, 0, FL_OUTCHAR, NULL },
                /* BRAND_GENERIC_8 */
        {13, 62,  6, 2, FL_OUTINT, NULL },
        /* I_PRICE_8 */
        {13, 71,  7, 2, FL_OUTINT, NULL },
        /* OL_AMOUNT_8 */

        {14,  2,  4, 0, FL_BINSHORT, NULL },
                /* OL_SUPPLY_W_ID_9 */
        {14,  9,  6, 0, FL_BININT, NULL },
        /* OL_I_ID_9 */
        {14, 18, 24, 0, FL_OUTSTR, NULL },
                /* I_NAME_9 */
        {14, 44,  2, 0, FL_INSHORT, NULL },
                /* OL_QUANTITY_9 */
        {14, 50,  3, 0, FL_OUTINT, NULL },
        /* S_QUANTITY_9 */
        {14, 57,  1, 0, FL_OUTCHAR, NULL },
                /* BRAND_GENERIC_9 */
        {14, 62,  6, 2, FL_OUTINT, NULL },
        /* I_PRICE_9 */
        {14, 71,  7, 2, FL_OUTINT, NULL },
        /* OL_AMOUNT_9 */

        {15,  2,  4, 0, FL_BINSHORT, NULL },
                /* OL_SUPPLY_W_ID_10 */
        {15,  9,  6, 0, FL_BININT, NULL },
        /* OL_I_ID_10 */
        {15, 18, 24, 0, FL_OUTSTR, NULL },
                /* I_NAME_10 */
        {15, 44,  2, 0, FL_INSHORT, NULL },
                /* OL_QUANTITY_10 */
        {15, 50,  3, 0, FL_OUTINT, NULL },
        /* S_QUANTITY_10 */
        {15, 57,  1, 0, FL_OUTCHAR, NULL },
                /* BRAND_GENERIC_10 */
        {15, 62,  6, 2, FL_OUTINT, NULL },
        /* I_PRICE_10 */
        {15, 71,  7, 2, FL_OUTINT, NULL },
        /* OL_AMOUNT_10 */

        {16,  2,  4, 0, FL_BINSHORT, NULL },
                /* OL_SUPPLY_W_ID_11 */
        {16,  9,  6, 0, FL_BININT, NULL },
        /* OL_I_ID_11 */
        {16, 18, 24, 0, FL_OUTSTR, NULL },
                /* I_NAME_11 */
        {16, 44,  2, 0, FL_INSHORT, NULL },
                /* OL_QUANTITY_11 */
        {16, 50,  3, 0, FL_OUTINT, NULL },
        /* S_QUANTITY_11 */
        {16, 57,  1, 0, FL_OUTCHAR, NULL },
                /* BRAND_GENERIC_11 */
```

```
{ 16, 62,  6, 2, FL_OUTINT, NULL },
/* l_PRICE_11 */
{ 16, 71,  7, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_11 */

{ 17,  2,  4, 0, FL_BINSHORT, NULL },
        /* OL_SUPPLY_W_ID_12 */
{ 17,  9,  6, 0, FL_BININT, NULL },
/* OL_I_ID_12 */
{ 17, 18, 24, 0, FL_OUTSTR, NULL },
        /* l_NAME_12 */
{ 17, 44,  2, 0, FL_INSHORT, NULL },
        /* OL_QUANTITY_12 */
{ 17, 50,  3, 0, FL_OUTINT, NULL },
/* S_QUANTITY_12 */
{ 17, 57,  1, 0, FL_OUTCHAR, NULL },
        /* BRAND_GENERIC_12 */
{ 17, 62,  6, 2, FL_OUTINT, NULL },
/* l_PRICE_12 */
{ 17, 71,  7, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_12 */

{ 18,  2,  4, 0, FL_BINSHORT, NULL },
        /* OL_SUPPLY_W_ID_13 */
{ 18,  9,  6, 0, FL_BININT, NULL },
/* OL_I_ID_13 */
{ 18, 18, 24, 0, FL_OUTSTR, NULL },
        /* l_NAME_13 */
{ 18, 44,  2, 0, FL_INSHORT, NULL },
        /* OL_QUANTITY_13 */
{ 18, 50,  3, 0, FL_OUTINT, NULL },
/* S_QUANTITY_13 */
{ 18, 57,  1, 0, FL_OUTCHAR, NULL },
        /* BRAND_GENERIC_13 */
{ 18, 62,  6, 2, FL_OUTINT, NULL },
/* l_PRICE_13 */
{ 18, 71,  7, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_13 */

{ 19,  2,  4, 0, FL_BINSHORT, NULL },
        /* OL_SUPPLY_W_ID_14 */
{ 19,  9,  6, 0, FL_BININT, NULL },
/* OL_I_ID_14 */
{ 19, 18, 24, 0, FL_OUTSTR, NULL },
        /* l_NAME_14 */
{ 19, 44,  2, 0, FL_INSHORT, NULL },
        /* OL_QUANTITY_14 */
{ 19, 50,  3, 0, FL_OUTINT, NULL },
/* S_QUANTITY_14 */
{ 19, 57,  1, 0, FL_OUTCHAR, NULL },
        /* BRAND_GENERIC_14 */
{ 19, 62,  6, 2, FL_OUTINT, NULL },
/* l_PRICE_14 */
{ 19, 71,  7, 2, FL_OUTINT, NULL },
/* OL_AMOUNT_14 */

{ 20,  2,  4, 0, FL_BINSHORT, NULL },
        /* OL_SUPPLY_W_ID_15 */
{ 20,  9,  6, 0, FL_BININT, NULL },
/* OL_I_ID_15 */
{ 20, 18, 24, 0, FL_OUTSTR, NULL },
        /* l_NAME_15 */
{ 20, 44,  2, 0, FL_INSHORT, NULL },
        /* OL_QUANTITY_15 */
{ 20, 50,  3, 0, FL_OUTINT, NULL },
/* S_QUANTITY_15 */
{ 20, 57,  1, 0, FL_OUTCHAR, NULL },
        /* BRAND_GENERIC_15 */
        { 20, 62,  6, 2, FL_OUTINT, NULL },
        /* l_PRICE_15 */
        { 20, 71,  7, 2, FL_OUTINT, NULL },
        /* OL_AMOUNT_15 */

        { 21, 18, 24, 0, FL_OUTSTR, NULL },
                /* Execution Status */
        { 21, 70,  8, 2, FL_OUTINT, NULL },
        /* TOTAL_AMOUNT */
        { -1, -1,  0, 0, 0, NULL },
};

const int       Start_field[6] = { 0, 1, 2, 1, 1, 2, };

********** makefile.fml **********
#
#               Makefile for test
#
#               Version     Beta2       1995/03/14
#               Version     1.0         1998/02/24 for
Solaris 2.x
#               Version     1.1         1998/06/01 for
Tc6/FML
#               Version     1.11        1998/06/12 add
term.c/o entry (^^;
#

ROOTDIR   = /opt/FSUNtpbs
# ROOTDIR = /opt/uxptuxt
RM          = rm -f
CC          = /opt/FSUNf90/bin/fcc
# CC        = /usr/ccs/bin/cc
# CC        = /usr/local/bin/gcc
# LIBS      = -lcurses
# LIBS      = -lncurses
MKFLDHDR            = $(ROOTDIR)/bin/mkfldhdr

INCLUDEDIR          = -I. -I$(ROOTDIR)/include
CCFLAGS             = $(INCLUDEDIR) -O -K 4 -K
ULTRA -DDUR -DSOLARIS2 -DUSE_FML -DTERM # -
DAVOID_RTE_BUG
# CCFLAGS                           = $(INCLUDEDIR)
-g -DDUR -DSOLARIS2 -DUSE_FML -DTERM -
DDEBUG=40
# LD_LIBRARY_PATH=$(ROOTDIR)/lib:/usr/ucb

all             : Tc

Tc              : Tc.c frame.o layout.o ui.o term.o dummy.o
fldtbl.h
                $(ROOTDIR)/bin/buildclient -v -o Tc -f "
$(CCFLAGS) Tc.c frame.o layout.o ui.o term.o
dummy.o "

fldtbl.h: fldtbl
                $(MKFLDHDR) fldtbl

term.o: term.c
                $(CC) -c $(CCFLAGS) term.c

layout.o: layout.c
                $(CC) -c $(CCFLAGS) layout.c

frame.o: frame.c
                $(CC) -c $(CCFLAGS) frame.c

ui.o            : ui.c
                $(CC) -c $(CCFLAGS) ui.c

dummy.o     : dummy.c
                $(CC) -c $(CCFLAGS) dummy.c

version.o : version.c
                -$(RM) version.c
                echo '#define DATE "\c' > version.c
                date >> version.c
                echo '"' >> version.c

clean           :
                -$(RM) Tc *.o fldtbl.h

********** misc.h **********
/*
                misc.h :

                Version     1.00        1996/12/26
                Version     1.01        1996/12/28
                ---------------------------
                Version     1.02        1997/05/20 for
term.c
                Version     1.03        1997/08/12 add
debug5()
                Version     1.03a       1998/06/13
                change comments
*/


#ifndef _MISC_H_
#define _MISC_H_

#include <stdio.h>
#include <ermo.h>

#if !defined( TRUE ) && !defined( FALSE )
# define      FALSE       0
# define      TRUE        1
#endif


#ifdef DEBUG
# define      debug( s )    fprintf s
# if ( DEBUG >= 10 )
#  define     debug1( s )   fprintf s
#  if ( DEBUG >= 20 )
#   define    debug2( s )   fprintf s
#   if ( DEBUG >= 30 )
#    define   debug3( s )   fprintf s
#    if ( DEBUG >= 40 )
#     define  debug4( s )   fprintf s
#     if ( DEBUG >= 50 )
#      define debug5( s ) fprintf s
#     endif
#    endif
#   endif
#  endif
# endif
#endif

#ifndef debug
# define      debug( s )
#endif
#ifndef debug1
# define      debug1( s )
#endif
#ifndef debug2
# define      debug2( s )
#endif
#ifndef debug3
```

```
# define       debug3( s )
#endif
#ifndef debug4
# define       debug4( s )
#endif
#ifndef debug5
# define       debug5( s )
#endif

#endif       /* _MISC_H_ */

********** Tc.c **********
/*
 *          Tc.c : main code for Tc6 / SymfoWARE
 *
 *          Version     0.90        1998/06/01 for
Tc6/FML
 *          Version     0.91        1998/06/03 delete
unused var., adjust const.
 *          Version     1.00        1998/06/05 delete
some #include directive
 *          Version     1.01        1998/06/11 add
ver. str.
 *          Version     1.02        1998/06/11 bug
fix for Tx_name[]
 *          Version     1.03        1998/06/12 add
write_errormessage(), message
 *
            entry for Anyerror()
 *          Version     1.04        1998/06/13 add
USE_TUXEDOLOG
 *          Version     1.04a       1998/06/13
            change debug level
 *          Version     1.05        1998/06/16 delete
init_tux(), clean_tux()
 *          Version     1.06        1998/06/17
            change messages/interface in Anyerror()
 *          Version     1.07        1998/06/18
            change SVC name for Delivery/Stocklevel
 *
            and declaration of Tx_name
 *          Version     1.08        1998/06/18 add
TPNOTIME for tpcall()/tpacall()
 *          Version     1.09        1998/06/19
            change Anyerror to sqlerror()
 *
            cast Trans_buf in tux_call()
 *          Version     1.10        1998/06/19 delete
Svc_num
 *          Version     1.11        1998/06/20 use
getenv() for Stocklevel/Delivery
 *
            service name;
 *          Version     1.15        1998/06/26 add
rangeerror() for carrier-id check
 *
            change sqlerror() message location,
 *
            Tstatus() interface for sqlerror()
 */

#include    <stdio.h>
#include    <stdlib.h>
#include    <string.h>
#include    <sys/times.h>
#include    <sys/time.h>
#include    <sys/param.h>
#include    <sys/ipc.h>
#include    <sys/msg.h>
#include    <math.h>
#include    <unistd.h>
#include    <signal.h>

#include    "misc.h"
#include    "bench2.h"
#include    "Tc.h"

#ifdef      SCRTEST
# include   "dummy.h"
#else
# include   "atmi.h"
# ifdef     USE_FML
#  include   "fml.h"
#  include   "fldtbl.h"
# endif
#endif


/* global variables */

const static char           _v_r[] = "Tc b8 d";
const char   *Tx_name[TX_NUM] = {
            "", "NewOrder", "Payment", "OrderStatus",
"Delivery", "StockLevel"
};

char        *Trans_buf;

int         W_id;
int         D_id;
int         T_id;

char        ServiceName[TX_NUM][SVC_LEN];


/*
            function prototypes
*/

void neworder_screen();
void payment_screen();
void orderstatus_screen();
void delivery_screen();
void stocklevel_screen();


/*
            set tuxedo service name
*/

void set_service_name()
{
            int         num;
            int         svrnum;
            int         svrnum_dl;
            int         svrnum_sl;
            char        *envptr;

            debug1( ( stderr, "IN:\t
set_service_name()\n" ) );
#ifdef      USE_FML
            sprintf( ServiceName[TX_NEWORDER],
            "TPCC" );
            sprintf( ServiceName[TX_PAYMENT],
            "TPCC" );
            sprintf(
ServiceName[TX_ORDERSTATUS],"TPCC" );
            sprintf( ServiceName[TX_DELIVERY],
            "TPCC" );
            sprintf( ServiceName[TX_STOCKLEVEL],
            "TPCC" );

            if ( ( envptr = getenv( "DEL_SVC" ) ) !=
NULL ) {
                        sprintf(
ServiceName[TX_DELIVERY], envptr );
            }
            if ( ( envptr = getenv( "STOCK_SVC" ) ) !=
NULL ) {
                        sprintf(
ServiceName[TX_STOCKLEVEL], envptr );
            }
#else
            num = ( T_id - 1 ) % 3200 + 1;
            svrnum    = ( num - 1 ) / 100 + 1;
            svrnum_dl = ( num - 1 ) / 800 + 1;
            svrnum_sl = 1;
            debug4( ( stderr, "num: %d, svrnum: %d,
svrnum_dl: %d, svrnum_sl: %d\n",
                        num, svrnum, svrnum_dl,
svrnum_sl ) );

            sprintf( ServiceName[TX_NEWORDER],
            "TPCC%d", svrnum );
            sprintf( ServiceName[TX_PAYMENT],
            "TPCC%d", svrnum );
            sprintf(
ServiceName[TX_ORDERSTATUS],"TPCC%d",
svrnum );
            sprintf( ServiceName[TX_DELIVERY],
            "TPCCd%d", svrnum );
            sprintf( ServiceName[TX_STOCKLEVEL],
            "TPCCs%d", svrnum );
#endif
            debug4( ( stderr, "\t ServiceName:
Neworder: %s, Payment: %s,\n"
                        "\t Orderstatus: %s, Delivery:
%s, Stocklevel: %s\n",

            ServiceName[TX_NEWORDER],
ServiceName[TX_PAYMENT],

            ServiceName[TX_ORDERSTATUS],
ServiceName[TX_DELIVERY],

            ServiceName[TX_STOCKLEVEL] ) );
            debug1( ( stderr, "OUT:\t
set_service_name()\n" ) );
}


/*
            main routine
*/

void TPCframe()
{
            bool        exitflag = FALSE;
            int         c;

            debug1( ( stderr, "IN:\t TPCframe()\n" ) );

            set_service_name();
            init_screen();

            while ( !exitflag ) {
```

```
                Tstatus( "D:Delivery S:Stock Level P:Payment O:Order Status"
                        " N:New Order Q:Quit", 1 );
                refresh();

                c = getch();

                switch ( c ) {
                case 'n':
                case 'N':
                        clear();
                        /* change_status( "New Order" ); */

        neworder_screen();
                        break;

                case 'p':
                case 'P':
                        clear();
                        /* change_status( "Payment" ); */
                        payment_screen();
                        break;

                case 'o':
                case 'O':
                        clear();
                        /* change_status( "Order Status" ); */

        orderstatus_screen();
                        break;

                case 'd':
                case 'D':
                        clear();
                        /* change_status( "Delivery" ); */
                        delivery_screen();
                        break;

                case 's':
                case 'S':
                        clear();
                        /* change_status( "Stock Level" ); */

        stocklevel_screen();
                        break;

                case 'q':
                case 'Q':
                        exitflag = TRUE;
                        break;

                default:
                        debug( ( stderr, "Illegal Select : '%c'\n", c ) );
                        break;
                }
        }

        close_screen();
        debug1( ( stderr, "OUT:\t TPCframe()\n" ) );
}
```

```
/*
                show status
*/

void Tstatus( char *status, int line )
{
        int     len = strlen( status );

        debug1( ( stderr, "IN:\t Tstatus( status: %s, line: %d )\n",
                                status, line ) );
                debug5( ( stderr, "\t LINES / COLS = %d / %d\n", LINES, COLS ) );

                if ( line > LINES || line < 1 ) {
                debug( ( stderr, "line (%d) is out of range [0:%d]\n",
                                                line, LINES ) );
                                line = 1;
                }
        move( LINES-line , (COLS/2)-(len/2) );
        debug5( ( stderr, "\t len: %d\n", len ) );
        deleteln();
        addstr( status );

        debug1( ( stderr, "OUT:\t Tstatus()\n" ) );
}


void change_status( char *s )
{
        char    buf[128];

        debug1( ( stderr, "IN:\t change_status( s: %s )\n", s ) );

        sprintf( buf, "%s screen...Use arrow keys to move "
                        "... Enter data in fields", s );
        Tstatus( buf, 1 );

        debug1( ( stderr, "OUT:\t change_status()\n" ) );
}


void errorstatus( char *s )
{
        char buf[128];

        debug1( ( stderr, "IN:\t errorstatus( s: %s )\n", s ) );

        sprintf( buf, "%s screen...Insufficient data ... "
                        "Enter data in fields", s );
        Tstatus( buf, 1 );
                refresh();
        debug( ( stderr, "Insufficient data in: %s\n", s ) );

        debug1( ( stderr, "OUT:\t errorstatus()\n" ) );
}


void rangeerror( char *s )
{
        char buf[128];

        debug1( ( stderr, "IN:\t rangeerror( s: %s )\n", s ) );

        sprintf( buf, "%s screen ... out of range", s );
        Tstatus( buf, 1 );
                refresh();
        debug( ( stderr, "Insufficient data in: %s\n", s ) );

        debug1( ( stderr, "OUT:\t rangeerror()\n" ) );
}


void sqlerror( int tx_type, char *bp )
{
                const char   *sqlfunc[] = {
                                        "SQLERROR occurred",
                                        "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",
                                };
                int             errorpos = *( ( int * )( bp + 8 ) );
                int             sqlstate = *( ( int * )( bp + 12 ) );
                int             pos;
                int             rtn;
                char            buf[4096];

        debug1( ( stderr, "IN:\t sqlerror( errorpos: %d, sqlstate: %d )\n",
                                errorpos, sqlstate ) );

                pos = errorpos / 100;
                if ( pos < 0 || pos > 4 ) {
                                pos = 0;
                }

                rtn = sprintf( buf, "%s ... ( SQLSTATE : %05d )",
                                sqlfunc[pos], sqlstate );
                Tstatus( buf, 2 );
                refresh();

                buf[rtn] = '\n';
                buf[rtn+1] = '\t';
                buf[rtn+2] = ' ';
                make_errorinfo( buf + rtn + 3, tx_type, bp );
                write_errormessage( buf );

                rtn = getch();
                debug4( ( stderr, "\t rtn key: %d\n", rtn ) );

        debug1( ( stderr, "OUT:\t sqlerror()\n" ) );
}


/*
                convert date/time
*/

void convert_datetime( char *out, double t )
{
                struct tm       tim;
                time_t                  tt = ( time_t )t;

                debug1( ( stderr, "IN:\t convert_datetime( out: %s, t: %f )\n",
                                out, t ) );
```

```
	tim = *( localtime( &tt ) );
	sprintf( out, "%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 );

	debug1( ( stderr, "OUT:\t convert_datetime( out: %s )\n", out ) );
}


void convert_date( char *out, double t )
{
	struct tm	tim;
	time_t		tt = ( time_t )t;

	debug1( ( stderr, "IN:\t convert_date( out: %s, t: %f )\n", out, t ) );

	tim = *( localtime( &tt ) );
	sprintf( out, "%02d-%02d-%04d",
		tim.tm_mday, tim.tm_mon + 1, tim.tm_year +1900 );

	debug1( ( stderr, "OUT:\t convert_date( out: %s )\n", out ) );
}


/*
	Tuxedo call
*/

void make_errorinfo( char *buf, int tx_type, char *bp )
{
	neworder_trans		*nt = ( neworder_trans * )bp;
	payment_trans		*pt = ( payment_trans * )bp;
	orderstat_trans		*ot = ( orderstat_trans * )bp;
	delivery_trans		*dt = ( delivery_trans * )bp;
	stocklvl_trans		*st = ( stocklvl_trans * )bp;
	int			i;
	char			tmp[128];

	sprintf( buf, "%s error: tperrno: %d, C_R: %d, "
		"SVC = '%s', pid: %d, tid: %d\n", Tx_name[tx_type], tperrno,
		*((int *)(bp+4)), ServiceName[tx_type], getpid(), T_id );

	switch ( tx_type ) {
	case TX_NEWORDER:
		sprintf( tmp, "\t w_id: %d, d_id: %d, c_id: %d, o_ol_cnt: %d\n",
			nt->w_id, nt->d_id, nt->c_id, nt->o_ol_cnt );
		strcat( buf, tmp );
		for ( i = 0; i < nt->o_ol_cnt; i++ ) {
			sprintf( tmp, "\t %02d: ol_w_id: %d, ol_quantity: %d,"
				"ol_i_id: %d\n", i+1, nt->ol_supply_w_id[i],
				nt->ol_quantity[i], nt->ol_i_id[i] );
			strcat( buf, tmp );
		}
		break;
	case TX_PAYMENT:
		sprintf( tmp, "\t w_id: %d, d_id: %d, c_id: %d, h_amount: %d\n"
			"\t c_w_id: %d, c_d_id: %d, c_last: %s\n",
			pt->w_id, pt->d_id, pt->c_id, pt->h_amount,
			pt->c_w_id, pt->c_d_id, pt->c_last );
		strcat( buf, tmp );
		break;
	case TX_ORDERSTATUS:
		sprintf( tmp, "\t w_id: %d, d_id: %d, c_id: %d, c_last: %s\n",
			ot->w_id, ot->d_id, ot->c_id, ot->c_last );
		strcat( buf, tmp );
		break;
	case TX_DELIVERY:
		sprintf( tmp, "\t w_id: %d, d_id: %d, o_carrier_id: %d\n",
			dt->w_id, dt->d_id, dt->o_carrier_id );
		strcat( buf, tmp );
		break;
	case TX_STOCKLEVEL:
		sprintf( tmp, "\t w_id: %d, d_id: %d, threshold: %d\n",
			st->w_id, st->d_id, st->threshold );
		strcat( buf, tmp );
		break;
	default:
		*buf = '\0';
	}
}


void tux_call( int tx_type, char **bp, long bpsize )
{
	int		rtn;
	long		rtnsize = 0;
	int		retrycnt = 0;

#ifdef	USE_FML
	Fchg( ( FBFR * )Trans_buf, FML_TERM, 0, (char *)&W_id, 0 );
	Fchg( ( FBFR * )Trans_buf, FML_TRAN, 0, (char *)&tx_type, 0 );
	Fchg( ( FBFR * )Trans_buf, FML_DATA, 0, *bp, ( FLDLEN )bpsize );
#else
	rtnsize = bpsize;
#endif

retry:
	if ( tx_type == TX_DELIVERY ) {
		rtn = tpacall( ServiceName[tx_type], Trans_buf, rtnsize,
			TPSIGRSTRT | TPNOREPLY | TPNOTIME );
	} else {
		rtn = tpcall( ServiceName[tx_type], Trans_buf, rtnsize,
			&Trans_buf, &rtnsize, 0 | TPNOTIME );
	}

	if ( rtn == -1 ) {
		char	buf[4096];

		make_errorinfo( buf, tx_type, *bp );

		switch ( tperrno ) {
		case TPETIME:
		/* timeout */
			if ( retrycnt++ > RETRY_COUNT ) {

		debug( ( stderr, "\t %s retry count : %d\n",

		Tx_name[tx_type], retrycnt ) );

		fatalerror( buf );
			}
			/* through (for retry) */

		case TPESVCERR:
		/* SVC communication error */
		case TPGOTSIG:
			debug( ( stderr, "\t Retry : %s\n", Tx_name[tx_type],
				buf ) );
			sleep( RETRY_INTERVAL );
			goto retry;
			break;

		case TPESVCFAIL:	/* SVC app detects error */
			debug( ( stderr, "\t %s : Service detects error\n",

		Tx_name[tx_type] ) );
			break;
		/* check in app. layer */

		default:
			fatalerror( buf );
			break;
		}
	}

	if ( tx_type != TX_DELIVERY ) {
#ifdef	USE_FML
		memcpy( *bp, Ffind( ( FBFR * )Trans_buf, FML_DATA, 0, NULL ),
			bpsize );
#else
		*bp = Trans_buf;
#endif
	}
}


/*
	NewOrder screen
*/
```

```
void neworder_screen()
{
#ifdef	USE_FML
	neworder_trans		tbuf;
	neworder_trans		*bp =
&tbuf;
#else
	neworder_trans		*bp =
( neworder_trans * )Trans_buf;
#endif

	debug1( ( stderr, "IN:\t
neworder_screen()\n" ) );

	query_neworder_frame( bp );
	tux_call( TX_NEWORDER, ( char ** )&bp,
sizeof( neworder_trans ) );
	display_neworder_frame( bp );

	debug1( ( stderr, "OUT:\t
neworder_screen()\n" ) );
}


/*
	Payment screen
*/

void payment_screen()
{
#ifdef	USE_FML
	payment_trans		tbuf;
	payment_trans		*bp =
&tbuf;
#else
	payment_trans		*bp =
( payment_trans * )Trans_buf;
#endif

	debug1( ( stderr, "IN:\t
payment_screen()\n" ) );

	query_payment_frame( bp );
	tux_call( TX_PAYMENT, ( char ** )&bp,
sizeof( payment_trans ) );
	display_payment_frame( bp );

	debug1( ( stderr, "OUT:\t
payment_screen()\n" ) );
}


/*
	Order status screen
*/

void orderstatus_screen()
{
#ifdef	USE_FML
	orderstat_trans		tbuf;
	orderstat_trans		*bp =
&tbuf;
#else
	orderstat_trans		*bp =
( orderstat_trans * )Trans_buf;
#endif

	debug1( ( stderr, "IN:\t
orderstatus_screen()\n" ) );

	query_orderstatus_frame( bp );
	tux_call( TX_ORDERSTATUS, ( char **
)&bp, sizeof( orderstat_trans ) );
	display_orderstatus_frame( bp );

	debug1( ( stderr, "OUT:\t
orderstatus_screen()\n" ) );
}


/*
	Delivery screen
*/

void delivery_screen()
{
#ifdef	USE_FML
	delivery_trans		tbuf;
	delivery_trans		*bp =
&tbuf;
#else
	delivery_trans		*bp =
( delivery_trans * )Trans_buf;
#endif

	debug1( ( stderr, "IN:\t delivery_screen()\n"
) );

	query_delivery_frame( bp );
	usleep( 1000 * 300 );		/*
98.7.21 add for remote audit */
	tux_call( TX_DELIVERY, ( char ** )&bp,
sizeof( delivery_trans ) );
	display_delivery_frame( bp );

	debug1( ( stderr, "OUT:\t
delivery_screen()\n" ) );
}


/*
	Stock Level screen
*/

void stocklevel_screen()
{
#ifdef	USE_FML
	stocklvl_trans		tbuf;
	stocklvl_trans		*bp =
&tbuf;
#else
	stocklvl_trans		*bp =
( stocklvl_trans * )Trans_buf;
#endif
	debug1( ( stderr, "IN:\t
stocklevel_screen()\n" ) );

	query_stocklevel_frame( bp );
	tux_call( TX_STOCKLEVEL, ( char ** )&bp,
sizeof( stocklvl_trans ) );
	display_stocklevel_frame( bp );

	debug1( ( stderr, "OUT:\t
stocklevel_screen()\n" ) );
}


/* error logging */

void write_errormessage( char *msg )
{
	FILE	*err;
	char	path[32];
	time_t	t;

#ifdef	USE_TUXEDOLOG
	userlog( "(W:%d D:%d)\n%s\n", W_id,
D_id, msg );
#else
	sprintf( path, "/tmp/tcerror.%05d", T_id );
	if ( ( err = fopen( path, "a+" ) ) != NULL ) {
		time( &t );
		fprintf( err, "%s(W:%d D:%d
PID:%d)\n",
			ctime( &t ), W_id,
D_id, getpid() );
		fputs( msg, err );
		fclose( err );
	}
#endif
}


/* Close screen and print the fatal error message to
stderr */

void fatalerror( char *msg )
{
	tpterm();
	close_screen();
	write_errormessage( msg );
	exit( -1 );
}


/* Signal handler */

void interrupt( int sig )
{
	if ( sig == SIGHUP ) {		/*
in.telnetd send SIGHUP */
		exit( -10 );
	} else {
		char	buf[1024];

		sprintf( buf, "Signal received :
sig = %d\n", sig );
		fatalerror( buf );
	}
}


/* startup routine */

int main( int argc, char *argv[] )
{
	/* initialize global variables */

	if ( argc < 2 ) {
		fprintf( stderr, "Argument
error!\n" );
		exit( 1 );
	}

	T_id = atoi( argv[1] );
	W_id = (T_id-1)/10 + 1;
	D_id = (T_id-1)%10 + 1;
```

```
        /* register interrupt handler */

        signal( SIGHUP, interrupt );
        signal( SIGINT, interrupt );
        signal( SIGTERM, interrupt );

        /* tuxedo setup */

        if ( tpinit( NULL ) == -1 ) {
                fprintf( stderr, "Failed to
connect service.\n" );
                exit( 1 );
        }

#ifdef  USE_FML
        if ( ( Trans_buf = (void *)tpalloc( "FML",
NULL, TRANS_SIZE ) )
                == NULL )
        {
                fprintf( stderr, "tpalloc()
failed.\n" );
                exit( 1 );
        }
#else
        if ( ( Trans_buf = (void *)tpalloc(
"CARRAY", NULL, TRANS_SIZE ) )
                == NULL )
        {
                fprintf( stderr, "tpalloc()
failed.\n" );
                exit( 1 );
        }
        memset( Trans_buf, 0, TRANS_SIZE );
#endif

#ifdef  DEBUG
        {
                char    buf[32];
                sprintf( buf,
"/tmp/tcheck.%05d", T_id );
                freopen( buf, "w", stderr );
                setvbuf( stderr, NULL, _IOLBF,
0 );
        }
#else
        fclose( stderr );                       /*
black magic */
#endif


        TPCframe();
        /* main loop */
        tpterm();

        return 0;
}

********** Tc.h **********
/*
 *      Tc.h : header file for Tc.c
 *
 *      Version 1.00    1998/06/02 for
Tc6/FML
 *      Version 1.01    1998/06/12 adjust
RETRY_ param.
 *      Version 1.02    1998/06/18
        change declaration of Tx_name
 *      Version 1.03    1998/06/19
        change Anyerror() to sqlerror()
 *
        add declaration of make_errorinfo(),
 *
        write_errormessage()
 *      Version 1.04    1998/06/19
        Svc_num deleted
 *      Version 1.05    1998/06/26
        change CHECKOK for range check,
 *
        change Tstatus() interface
 */

#ifndef _TC_H_

#define _TC_H_

#include "ui.h"

#define INTNULL             -32768

#define TX_NEWORDER         1
#define TX_PAYMENT          2
#define TX_ORDERSTATUS      3
#define TX_DELIVERY         4
#define TX_STOCKLEVEL       5
#define TX_NUM              (5+1)

#define SVC_LEN             16
#define TRANS_SIZE          1264
#define RETRY_INTERVAL      2
#define RETRY_COUNT         0x7FFFFFFF

#define CHECKOK             0x80000000
#define NOERR               1

extern char *Trans_buf;

extern int  W_id;
extern int  D_id;
extern int  T_id;

extern char ServiceName[TX_NUM][SVC_LEN];
extern const char       *Tx_name[TX_NUM];


/* function prototype */

void Tstatus( char *, int );
void errorstatus( char * );
void change_status( char * );
void make_errorinfo( char *, int, char * );
void write_errormessage( char * );
void fatalerror( char * );
void sqlerror( int, char * );
void convert_datetime( char *, double );
void convert_date( char *, double );
void tux_call( int, char **, long );
void init_tux();
void clean_tux();

#endif

********** term.c **********
/*
        term.c  :

        Version 0.50    1997/05/16
        Research version
        Version 0.60    1997/05/20 for
Tc.2
        Version 0.70    1997/05/21 for
Tc.4 (-->Tc4.std)
        Version 0.90    1997/07/17 ESC
analyze
        Version 0.91    1997/07/30 ESC
improve
        Version 0.92    1998/05/27 for
Tc6/SymfoWARE/Solaris2.x
        Version 0.94    1998/06/05 delete
w*() functions, struct WINDOW
        Version 0.95    1998/06/09
        rewrite getch()/get_ESC()
        Version 0.96    1998/06/10 delete
not-implemented func.
        Version 0.97    1998/06/13 add
buffer dump for debug
        Version 0.97a   1998/06/13 adjust
debug level
*/

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <termios.h>

#include "term.h"
#include "misc.h"

#define BUFFERING
/*#define ASSIGN_BUFFER */

#ifdef  BUFFERING
# define WRITE(str, len)                fwrite(
(str), 1, (len), stdout )
# ifdef ASSIGN_BUFFER
#  define BUFFER_SIZE                   4096
# endif
#else
# define WRITE(str, len)                write(
STDOUT, (str), (len) )
#endif


int         LINES = 24;
            /* vt100 default */
int         COLS  = 80;
            /* vt100 default */
WINDOW      *stdscr = NULL;
            /* for compatibility */

static struct termios   orig_stat;
static int              bufrest = 0;            /* (^^;
*/
static int              _attr = A_NORMAL;
static short            _ccol = 0;
static short            _crow = 0;

/*
static char _writebuffer[4096];
static int  _writesize = 0;
*/


#if 0

/* internal function */
```

```
int itoa( char *out, int in )
{
        char    buf[16];
    char    *bufptr = buf + 16;
        int     len = 0;

    if ( in == 0 ) {
        *out = '0';
        *( out + 1 ) = '\0';
        return 1;
    } else if ( in < 0 ) {
                in = -in;
        *out++ = '-';
                len++;
    }

    *--bufptr = '\0';
    for ( ; in > 0; in /= 10 ) {
        *--bufptr = in % 10 + '0';
                len++;
    }

        while ( *out++ = *bufptr++ )
                ;

        return len;
}

#endif


/* buffering write */

/*
int bufflush()
{
        write( STDOUT, _writebuffer, _writesize )
!= _writesize );
        _writesize = 0;

        return 0;
}

int bufwrite( char *data, int size )
{
        if ( _writesize + size > 4096 ) {
                bufflush();
        }
        memcpy( _writebuffer + _writesize, data,
size );

        return size;
}

int bufputch( char ch )
{
        if ( _writesize + 1 > 4096 ) {
                buflush();
        }
        _writebuffer[_writesize++] = ch;

        return 0;
}
*/


/* new/destory window */

WINDOW *initscr()
{
        struct termios          stat;
        struct winsize          size;
#if defined(BUFFERING) &&
defined(ASSIGN_BUFFER)
        static char
        stream_buf[BUFFER_SIZE];

        setvbuf( stdout, stream_buf, _IOFBF,
BUFFER_SIZE );
#endif
#ifdef  BUFFERING
        setvbuf( stdout, NULL, _IOFBF, 4096 );
#endif

        if ( tcgetattr( 0, &orig_stat ) == 0 ) {

                stat = orig_stat;

                /* set raw mode */

                stat.c_iflag &=
~(INLCR|ICRNL|IUCLC|ISTRIP|IXON);
                stat.c_oflag &=
~(ONLCR|OCRNL|ONLRET|OPOST);
                stat.c_lflag &=
~(ICANON|ECHO);

#if 1
                stat.c_cc[VMIN]  = 1;
        /* read 1 char at least */
                stat.c_cc[VTIME] = 0;
        /* wait until char comes */
#else
                stat.c_cc[VMIN]  = 16;
        /* read 1 - 16 char(s) */
                stat.c_cc[VTIME] = 2;
        /* 2-tick wait after 1st char */
#endif

                tcsetattr( STDIN, TCSADRAIN,
&stat );        /* set after output */

                /* get window size */

                if ( ioctl( STDIN,
TIOCGWINSZ, &size ) == 0 ) {
                        LINES =
size.ws_row;
                        COLS  =
size.ws_col;
                }
        }

        _crow = 0;
        _ccol = 0;

        clear();
        return;
}

int endwin()
{
        /* set after output */

        if ( tcsetattr( 0, TCSAFLUSH, &orig_stat ) <
0 ) {
                return -1;
        }
        return 0;
}


/* move cursor */

/* if using DS/cc, use combination of constant
literal/number and pointer
   ex: 'char *buf = "\033[A"; WRITE( buf, 3 )' */

void cursor_up1()
        /* cuu1 */
{
        char    buf[] = "\033[A";

        WRITE( buf, sizeof( buf ) - 1 );
        /* _crow--; */
}


void cursor_up( int row )                       /* cuu
*/
{
        char    buf[8];

        sprintf( buf, "\033[%dA", row );
        WRITE( buf, strlen( buf ) );
        /* _crow -= row; */
}

void cursor_down1()
        /* cud1 */
{
        char    buf[] = "\n";

        WRITE( buf, sizeof( buf ) - 1 );
        /* _crow++; */
}

void cursor_down( int row )
        /* cud */
{
        char    buf[8];

        sprintf( buf, "\033[%dB", row );
        WRITE( buf, strlen( buf ) );
        /* _crow += row; */
}

void cursor_right1()
        /* cuf1 */
{
        char    buf[] = "\033[C";

        WRITE( buf, sizeof( buf ) - 1 );
        /* _ccol++; */
}

void cursor_right( int col )
        /* cuf */
{
        char    buf[8];

        sprintf( buf, "\033[%dC", col );
        WRITE( buf, strlen( buf ) );
        /* _ccol += col; */
}
```

```
void cursor_left1()
        /* cub1 */
{
        char        buf[] = "\b";

        WRITE( buf, sizeof( buf ) - 1 );
        /* _ccol--; */
}

void cursor_left( int col )                     /* cub
*/
{
        char        buf[8];

        sprintf( buf, "\033[%dD", col );
        WRITE( buf, strlen( buf ) );
        /* _ccol -= col; */
}

int move( int y, int x )                        /* csr
*/
{
        char        buf[16];
        int         len;

        debug1( ( stderr, "IN:\t move( y: %d x: %d
)\n", y, x ) );
#if 1
        len = sprintf( buf, "\033[%d;%dH", y+1, x+1
);
        WRITE( buf, len );
#else
        if ( y == _crow ) {
                if ( x > _ccol ) {
                        cursor_right( x -
_ccol );
                } else if ( x < _ccol ) {
                        cursor_left( _ccol -
x );
                }
        } elseif ( x == _ccol ) {
                if ( y > _crow ) {
                        cursor_down( y -
_crow );
                } else if ( x < _ccol ) {
                        cursor_up( _crow -
y );
                }
        } else {
                buf[0] = '\033';
                buf[1] = '[';
                len = 2;
                len = itoa( buf + len, y+1 );
                buf[len++] = ';';
                len += itoa( buf + len, x+1 );
                buf[len++] = 'H';
                WRITE( buf, len );
        }
#endif
/*
        _ccol = x;
        _crow = y;
*/
        debug1( ( stderr, "OUT:\t move()\n" ) );
        return 0;
}


/* clear window/line */

int clear()
        /* clear */
{
        char        buf[] = "\033[H\033[J";

        WRITE( buf, sizeof( buf ) - 1 );
/*
        _ccol = 0;
        _crow = 0;
*/
        return 0;
}

int deleteln()
                /* el, el1 */
{
        char        buf[] = "\033[K\033[1K";  /* el :
clear to end of line */

                /* el1 : clear to beginning */

                /*       of line */
        WRITE( buf, sizeof( buf ) - 1 );
        /* _ccol = 0; */

        return 0;
}


/* change attributes */

void normal_mode()
        /* sgr0 */
{
        char        buf[] = "\033[m\017";

        WRITE( buf, sizeof( buf ) - 1 );
}

void reverse_mode()
        /* rev */
{
        char        buf[] = "\033[7m";

        WRITE( buf, sizeof( buf ) - 1 );
}

void underline_mode_on()
        /* smul */
{
        char        buf[] = "\033[4m";

        WRITE( buf, sizeof( buf ) - 1 );
}

void underline_mode_off()
        /* rmul */
{
        char        buf[] = "\033[m";

        WRITE( buf, sizeof( buf ) - 1 );
}

int attrset( int attrs )
{
        if ( attrs != _attr ) {
                normal_mode();
                                /* A_NORMAL */
                _attr = 0;
                /* reset */
                if ( attrs & A_UNDERLINE ) {

        underline_mode_on();
                        _attr |=
A_UNDERLINE;
                }
                if ( attrs & A_STANDOUT ) {
                        reverse_mode();
                        _attr |=
A_STANDOUT;
                }
        }

        return 0;
}

/* print character/string */

void addnstr( char *str, int n )
{
        WRITE( str, n );
        /* _ccol += n; */
}

void addstr( char *str )
{
        WRITE( str, strlen( str ) );
/*
        int         l;

        WRITE( str, ( l = strlen( str ) ) );
        _ccol += l;
*/
}

void addch( char ch )
{
        WRITE( &ch, 1 );
        /* _ccol++; */
}


/* input characters */

#if 0

int getch()
{
        unsigned char       buf[32];
        int                 rtn;
        struct termios      stat;

        rtn = read( STDIN, buf, 32 );

        if ( rtn <= 0 ) {                       /*
error or closed connection ... */
                return ERR;
        }

        if ( buf[0] == 0x1b ) {                 /*
found ESC sequence header */

                /* if many chars are read
immediately, this is ESC sequence */
```

```
                if ( tcgetattr( STDIN, &stat ) != 0 ) {
                        /* I cannot know ESC sequence in BUFFER mode ... */
                        /* Sorry... */

                        return buf[0];
                }
                stat.c_cc[VTIME] = 1;
        /* wait 1 tick */
                tcsetattr( STDIN, TCSADRAIN, &stat );

                rtn = read( STDIN, buf, 32 );
        /* read many chars */

                tcgetattr( STDIN, &stat );
                stat.c_cc[VTIME] = 0;
        /* restore */
                tcsetattr( STDIN, TCSADRAIN, &stat );

                if ( rtn <= 0 ) {
        /* ESC key was pressed */
                        return 0x1b;
        /* return ESC code */
                }

                /* data exists : convert sequence into a code */

                rtn = 0;
                switch ( buf[0] ) {
                case '[' :
                        switch ( buf[1] ) {
                        case 'C' :
                                rtn = KEY_RIGHT;
                                break;
                        case 'A' :
                                rtn = KEY_UP;
                                break;
                        case 'B' :
                                rtn = KEY_DOWN;
                                break;
                        case 'D' :
                                rtn = KEY_LEFT;
                                break;
                        /* case '3' : */
                        case '4' :

        switch( buf[2] ) {
                                case '~' :

        rtn = KEY_DC;

        break;
                                }
                                break;
                        }
                        break;
                }

                return rtn;
        }

        return buf[0];
        /* a key was pressed */
}

#else   /* 1 */

#define KEYBUFSIZE      7

int get_ESC( unsigned char *buf, int *key )
{
        unsigned char   *ptr = buf;

        debug1( ( stderr, "IN:\t get_ESC()\n" ) );
        switch ( *ptr ) {
        case '[':
                switch( *++ptr ) {
                case '4':
                        switch( *++ptr ) {
                        case '~':
                                *key = KEY_DC;  break;
                        case '\0':
                                return -1;      break;
                        default:
                                *key = ESC;     return 0;
                        }
                        break;
                case 'A':
                        *key = KEY_UP;
                        break;
                case 'B':
                        *key = KEY_DOWN;        break;
                case 'C':
                        *key = KEY_RIGHT;       break;
                case 'D':
                        *key = KEY_LEFT;        break;
                case '\0':
                        return -1;
                        break;
                default:
                        *key = ESC;
                        return 0;
                }
                break;
        case '\0':
                return -1;              break;
        default:
                *key = ESC;             return 0;
        }

        debug1( ( stderr, "OUT:\t get_ESC() = %d\n", ptr - buf + 1 ) );
        return ptr - buf + 1;
}


int getch()
{
        static unsigned char
                buf[KEYBUFSIZE+1];
        static int      bufcur  = 0;
        int             rtn = 0;
        int             ch = 0;

        debug1( ( stderr, "IN:\t getch()\n" ) );

        if ( bufrest == 0 ) {
                bufrest = read( STDIN, buf, KEYBUFSIZE );
                debug4( ( stderr, "\t read count = %d\n", bufrest ) );
                if ( bufrest <= 0 ) {
        /* error or lost connection */
                        debug1( ( stderr, "OUT:\t getch( ERR )\n" ) );
                        return ERR;
                }
                bufcur = 0;
        }

#if defined( DEBUG ) && ( DEBUG >= 40 )
        {
                int     i = 0;
                fputc( '\t', stderr );
                for ( i = 0; i <= KEYBUFSIZE; i++ ) {
                        fprintf( stderr, " %02X", buf[i] );
                }
                fprintf( stderr, "\t c: %d r: %d\n", bufcur, bufrest );
        }
#endif

        if ( buf[bufcur] != ESC ) {
                bufrest--;
                debug1( ( stderr, "OUT:\t getch() = '%c':%d\n",
                        buf[bufcur], buf[bufcur] ) );
                return buf[bufcur++];
        }

        /* ESC sequence is found. */

        buf[bufcur+bufrest] = '\0';     /* sentinel at the end of data */
        bufcur++;                       /* ESC has been read */
        bufrest--;

        rtn = get_ESC( buf+bufcur, &ch );

        if ( rtn < 0 ) {                /* reload */
                memmove( buf, buf+bufcur, bufrest );
                bufcur = 0;

                rtn = read( STDIN, buf+bufrest, KEYBUFSIZE - bufrest );
                while ( rtn > 0 ) {
                        bufrest += rtn;
                        buf[bufrest] = '\0';
                        if ( ( rtn = get_ESC( buf, &ch ) ) >= 0 ) {
                                break;
                        }
```

```
                                        rtn = read( STDIN,
buf+bufrest, KEYBUFSIZE - bufrest );
                        }
                        if ( rtn < 0 ) {/* connection error
*/
                                        debug1( ( stderr,
"OUT:\t getch() = ERR\n" ) );
                                        return ERR;
                        } else if ( rtn == 0 ) {        /*
buffer full or ESC */
                                        debug1( ( stderr,
"OUT:\t getch() = ESC\n" ) );
                                        return ESC;
                        }
        }

        bufrest -= rtn;
        bufcur  += rtn;

        debug4( ( stderr, "\t bufcur = %d, bufrest =
%d, rtn = %d\n",
                        bufcur, bufrest, rtn ) );
        debug1( ( stderr, "OUT:\t getch() = %d\n",
ch ) );
        return ch;
}

#endif  /* 1 */


/* update screen */

int doupdate()
{
#ifdef  BUFFERING
        if ( bufrest == 0 ) {                   /* for
write buffering */
                fflush( stdout );
        }
#endif
        return 0;
}

********** term.h  **********
/*
        term.h :

        Version 0.10    1997/05/20 First
Version (for Tc.2)
        Version 0.20    1997/05/21 add
function prototype
        Version 0.50    1997/07/03 for
Tc.4 (-->Tc4.std)
        Version 0.70    1997/07/16 ESC
analyze
        Version 0.71    1997/07/31
        0.70+0.50 (define getch())
        Version 0.72    1998/05/25 add
addch()
        Version 0.80a   1998/06/05 delete
w*() functions
        Version 0.81    1998/06/05 add
_TERM_H_ (^^;
        Version 0.82    1998/06/10 delete
not-implemented func.
*/

#ifndef _TERM_H_
#define _TERM_H_

#define STDIN   0
#define STDOUT  1
#define STDERR  2

#define EDEL            4
#define BS              8
#define TAB             9
#define LF              10
#define CR              13

#define CUU             16
#define CUD             14
#define CUF             6
#define CUB             2

#define ESC             033
#define DEL             0177

#define KEY_MIN         0401
#define KEY_UP          0401
#define KEY_DOWN        0402
#define KEY_LEFT 0403
#define KEY_RIGHT       0404

#define KEY_DC          0411
#define KEY_BACKSPACE   0412
#define KEY_PAUSE       0413

#define KEY_INS         0420
#define KEY_DEL         (KEY_DC)
#define KEY_HOME        0422
#define KEY_END         0423
#define KEY_PAGEUP 0424
#define KEY_PAGEDOWN    0425

#define KEY_UNKNOWN     0477

#define KEY_F0          0500
#define KEY_F(n)   (KEY_F0+(n))
#define KEY_PF0         0520
#define KEY_PF(n) (KEY_PF0+(n))
#define KEY_SF0         0540
#define KEY_SF(n) (KEY_PF0+(n))

#define KEY_MAX         0577

#define A_NORMAL        0
#define A_UNDERLINE     1
#define A_STANDOUT      4

#define ACS_HLINE       '-'
#define ACS_VLINE       '|'
#define ACS_LLCORNER    '+'
#define ACS_LRCORNER    '+'
#define ACS_ULCORNER    '+'
#define ACS_URCORNER    '+'

#ifndef FALSE
#define FALSE 0
#endif

#ifndef TRUE
#define TRUE 1
#endif

#ifndef ERR
#define ERR (-1)
#endif

#endif

#define WINDOW  void
        /* for compatibility */

typedef char    bool;

extern int  LINES, COLS;
extern WINDOW   *stdscr;                /* for
compatibility */


/* function prototype */

WINDOW *initscr();
int endwin();
void cursor_up1();
void cursor_up( int );
void cursor_down1();
void cursor_down( int );
void cursor_right1();
void cursor_right( int );
void cursor_left1();
void cursor_left( int );
void normal_mode();
void reverse_mode();
void underline_mode_on();
void underline_mode_off();
int clear();
int deleteln();
int move( int, int );
void addnstr( char *, int );
void addstr( char * );
void addch( char );
int getch();
int attrset( int attrs );
int doupdate();
int refresh();

/* macro */
/*#define   doupdate()          fflush( stdout ) */
#define refresh()               doupdate()
#define bkgd(a)
#define resetty()
#define savetty()
#define noecho()
#define cbreak()
#define intrflush(w,b)
#define keypad(w,b)
#define leaveok(w,b)
#define nodelay(w,b)
#define nonl()
#define slk_clear()

#endif  /* _TERM_H_ */

********** ui.c **********
/*
        ui.c : Module for field operations

        Version 1.00    1996/12/26 First
edition
        ----------
        Version 1.01    1997/01/29
        ----------
        Version 1.02    1997/05/19 for
term.c
        ----------
        Version 1.03    1997/07/30
```

```
                ---------------------------
                Version     2.00        1998/05/27
                Second edition (for Tc6)
                Version     2.11        1998/06/02
                display_frame() changed / DUR deleted
                Version     2.20        1998/06/02 omit
FL_BLANK / *_dirty in queries
                Version     2.21        1998/06/03 check
by isalnum()
                Version     2.21a       1998/06/13
                change debug level
*/

#include    <stdio.h>
#include    <stdlib.h>
#include    <string.h>
#include    "ui.h"
#include    "misc.h"


/* functions for editing a number-field */

/*                                      */

#if 0
void extract_number( char *buf, datafield_t *df )
{
                if ( df->flag & FL_INT ) {
                        sprintf( buf, "%d", *( ( int * )df-
>data ) );
                } else if ( df->flag & FL_DBL ) {
                        sprintf( buf, "%.0f", *( ( double *
)df->data ) );
                } else if ( df->flag & FL_SHORT ) {
                        sprintf( buf, "%d", *( ( short *
)df->data ) );
                }
}
#else
void extract_number( char *buf, datafield_t *df )
{
                int             i;
                char            *bufptr = buf + df->len;

                if ( df->flag & FL_DBL ) {
                        sprintf( buf, "%.0f", *( ( double *
)df->data ) );
                        return;
                }

                if ( df->flag & FL_SHORT ) {
                        i = *( ( short * )df->data );

                } else if ( df->flag & FL_INT ) {
                        i = *( ( int * )df->data );
                }

                if ( i == 0 ) {
                        *buf = '0';
                        *( buf + 1 ) = '\0';
                        return;
                } else if ( i < 0 ) {
                        *buf++ = '-';
                        i = -i;
                }

                *bufptr = '\0';
                for ( ; i > 0; i /= 10 ) {
                        *--bufptr = i % 10 + '0';
                }

                strcpy( buf, bufptr );
}
#endif


/*                                      */

int make_number_field( char *buf, datafield_t *df )
{
                char            nstr[32];               /* 32
(^^; */
                char            *nstrptr = nstr;
                                /*              */
                char            *bufend = buf + df->len;
                /*              */
                char            *bufptr = buf;
                                        /*              */
                int             srclen;
                int             ilen;
                int             len = df->len;
                                        /*              */

                *bufend = '\0';
                                        /*              */
                extract_number( nstr, df );
                        /*          */

                if ( df->flag & FL_SIGN ) {
                        if ( *nstrptr == '-' ) {
                /*          */
                                        *bufptr++ = '-';
                                /*              */
                                } else {
                                        *bufptr++ = ' ';
                                }
                                len--;
                }
                if ( *nstrptr == '-' ) {
                /*          */
                        nstrptr++;
                        /*          */
                }

                srclen = strlen( nstrptr );
                /*          */
                ilen = srclen - df->flen;
                /*          */


                if ( ( srclen > len ) || ( ( df->flen > 0 ) && (
srclen + 1 > len ) ) )
                {
                        /*

                                                */
                        memset( buf, '*', df->len );
                        debug( ( stderr, "\t '%s': out of
range for field (%d,%d)\n",
                                                nstr, df->row, df-
>col ) );
                        return 0;
                }

                memset( bufptr, ' ', len );
                /*          */

                if ( df->flen == 0 ) {
                        /*          */
                        bufend -= ilen;
                                        /*          */
                        memcpy( bufend, nstrptr, ilen );
                        len = ilen;
                } else if ( ilen <= 0 ) {
                /*          */
                        bufend -= df->flen;
                        /*              */
                        *( bufend - 1 ) = '.';
                        *( bufend - 2 ) = '0';
                        memset( bufend, '0', df->flen -
srclen );
                        memcpy( bufend + df->flen -
srclen, nstrptr, srclen );
                        len = 2 + df->flen;
                } else {
                        /*          */
                        bufend -= srclen + 1;
                        /*          */
                        memcpy( bufend, nstrptr, ilen );
                        *( bufend+ilen ) = '.';
                        memcpy( bufend+ilen+1,
nstrptr+ilen, df->flen );
                        len = ilen + 1 + df->flen;
                }

                return len;
                /*              */
}


/*                                      */

void insert_number( datafield_t *df, char *buf, int *len,
int ch, int *pos )
{
                int             fpos = df->len - df->flen - 1;
                int             start;
                int             end;
                int             i;
                char            tmp;

                /*                              */

                if ( ch == '-' ) {
                /*              */
                        if ( df->flag & FL_SIGN ) {
                                if ( *buf == '-' ) {
                        /* toggle switch */
                                        *buf =
' ';
                                } else {
                                        *buf =
'-';
                                }
                                move( df->row, df-
>col );
                                addch( *buf );
                                /*          */
                                move( df->row, df-
>col + *pos );

                                return;
                        } else {
                                return;
                                /*      */
                        }
                }
```

```
        if ( df->flag & FL_SIGN ) {
                if ( *len == df->len - 1 ) {
/*              */
                        return;
                }
        } else if ( *len == df->len ) {
                /*              */
                return;
        }

        end = *pos - 1;
        /*              (       ) */
        start = df->len - *len - 1;
        /*          */
        for ( i = start; i < end; i++ ) {
        /*          */
                buf[i] = buf[i+1];
        }
        buf[end] = ch;
                /*      */
        ( *len )++;
        /*              */

        /* '0'  '0.xxx'             '0x'  '0x.xxx'
          buf[start]      '0'     '0'
*/

        if ( buf[start] == '0' ) {
                buf[start++] = ' ';
        /* start moves to right */
                ( *len )--;
        }

        if ( ( df->flen > 0 ) && ( fpos <= end ) ) {
                /*


*/
                tmp = buf[fpos-1];
                buf[fpos-1] = buf[fpos];
                buf[fpos] = tmp;
        }

        move( df->row, df->col + start );
        addnstr( buf + start, *len );
/*
        move( df->row, df->col );
        addstr( buf );
*/
        move( df->row, df->col + *pos );
}


/*                              */

void delete_number( datafield_t *df, char *buf, int *len,
int *pos )
{
        int         fpos = df->len - df->flen - 1;
                    /*          */
        int         end;
        int         i;
        char        tmp;

        /*                      */

        if ( ( *len == 0 ) || ( *pos == df->len ) || (
buf[*pos] == '.' )
                || ( buf[*pos] == ' ' ) )
        {
                /*


                        */
                return;
        }

        end = df->len - *len;
                /*          */
        for ( i = *pos - 1; i >= end; i-- ) {
        /*          */
                buf[i+1] = buf[i];
        }
        buf[end] = ' ';
                /*              */
        ( *len )--;
        /*              */
        if ( buf[*pos] == ' ' ) {
                /* *pos                 '000.xxxx'
'0'
                                        '0'
                  *pos + 1
fpos - 1
                        df->len - 1
*/
                if ( df->flen > 0 ) {
                        end = fpos - 1;
                } else {
                        end = df->len - 1;
                }
                for ( i = *pos + 1; i <= end; i++ )
{
                        if ( buf[i] != '0' ) {
                                break;
                        }
                        buf[i] = ' ';
                        ( *len )--;
                }
                /* *len -= i - ( *pos + 1 ); */
                *pos = i - 1;               /*
*/
        }

        if ( df->flen > 0 ) {
        /*              */
                if ( fpos < *pos ) {
                        /*


                                        */
                        tmp = buf[fpos];
                        buf[fpos] =
buf[fpos+1];
                        buf[fpos+1] = tmp;
                }

                if ( buf[fpos-1] == ' ' ) {
                        /*
'.xxx'
                          '0'       */
                        buf[fpos-1] = '0';
                        ( *len )++;

                        /*
*/
                        if ( df->flag &
FL_SIGN ) {
                                for ( i
= df->flen; i > 0; i-- ) {
                if ( buf[fpos+i] != '0' ) {
                        break;
                }
                                }
                                if ( i
== 0 ) {
                *buf = ' ';
                                }
                        }
                } else {
                        /*              */
                        /*          == 0
                                  '0'           */
                        if ( *len == 0 ) {
                                if ( df->flag &
FL_SIGN ) {
                                        *buf =
' ';                    /*      */
                                }
                                buf[df->len - 1] =
'0';
                        }
                }

                if ( *len == 0 ) {
                        *pos = df->len;
                        /*                  */
                }

                move( df->row, df->col );
                addstr( buf );
                move( df->row, df->col + *pos );
}

/*                              */

void save_number( datafield_t *df, char *buf, int len )
{
        char        *bufptr = buf + df->len - len;
                    /*              */
        int         i = 0;
        short       s = 0;
        double      d = 0.0;

        if ( df->flag & FL_INT ) {
                for ( ; isdigit( *bufptr ); bufptr++
) {
                        i = i * 10 + ( *bufptr
) - '0';
                }
                if ( *bufptr == '.' ) {
                        for ( bufptr += 1;
isdigit( *bufptr ); bufptr++ ) {
                                i = i *
10 + ( *bufptr ) - '0';
                        }
                }
                if ( *buf == '-' ) {
                        i = -i;
                }
                *( ( int * )df->data ) = i;
```

```
        } else if ( df->flag & FL_DBL ) {
                for ( ; isdigit( *bufptr ); bufptr++
) {
                        d = d * 10 + (
*bufptr ) - '0';
                }
                if ( *bufptr == '.' ) {
                        for ( bufptr += 1;
isdigit( *bufptr ); bufptr++ ) {
                                d = d *
10 + ( *bufptr ) - '0';
                        }
                }
                if ( *buf == '-' ) {
                        d = -d;
                }
                *( ( double * )df->data ) = d;
        } else if ( df->flag & FL_SHORT ) {
                for ( ; isdigit( *bufptr ); bufptr++
) {
                        s = s * 10 + (
*bufptr ) - '0';
                }
                if ( *bufptr == '.' ) {
                        for ( bufptr += 1;
isdigit( *bufptr ); bufptr++ ) {
                                s = s *
10 + ( *bufptr ) - '0';
                        }
                }
                if ( *buf == '-' ) {
                        s = -s;
                }
                *( ( short * )df->data ) = s;
        }
}


/*                      */

int edit_number_field( datafield_t *df )
{
        int             exitflag = FALSE;
        int             ch;
        int             len;
        int             pos;
        int             start = 0;
        char            buf[64];
        char            nstr[64];

        len = make_number_field( buf, df );
        pos = df->len;
        if ( df->flag & FL_SIGN ) {
                start = 1;
        }

        move( df->row, df->col );
        attrset( ATTR_INPUT );
        addstr( buf );
        move( df->row, df->col + pos );
        refresh();

        while ( exitflag != TRUE ) {
                ch = getch();
                switch ( ch ) {
                case LF:
                case CR:
                case TAB:
                case KEY_UP:
                case CUU:
                case KEY_DOWN:
                case CUD:
                        exitflag = TRUE;
                        break;
                case KEY_LEFT:
                case CUB:
                        if ( ( pos > df->len
- len ) && ( pos != start ) ) {
                                pos--;

                        cursor_left1();
                        }
                        break;
                case KEY_RIGHT:
                case CUF:
                        if ( pos < df->len )
{

                        pos++;

                        cursor_right1();
                        }
                        break;
                case KEY_DC:
                case DEL:
                case EDEL:
                        delete_number(
df, buf, &len, &pos );
                        break;
                case BS:
                        /* BS =
KEY_LEFT + KEY_DEL + KEY_RIGHT (^^; */
                        if ( pos > start ) {
                                pos--;

                        cursor_left1();

                        delete_number( df, buf, &len, &pos );
                                if (
pos < df->len ) {

                        pos++;

                        cursor_right1();
                                }
                        }
                        break;
                default:
#ifdef  AVOID_RTE_BUG
                        if ( ch == ' ' ) {

                        getch();        /* skip char after space (--; */
                        }
#endif
                        if ( isdigit( ch ) || ch
== '-' ) {

                        insert_number( df, buf, &len, ch, &pos );
                        }
                        break;
                }
                refresh();
        }
        attrset( ATTR_BASE );
        save_number( df, buf, len );

        return ch;
}


/* functions for editign a string-field */

/*                                      */

int make_string_field( char *buf, datafield_t *df )
{
        int             srclen;
        int             rtn;

        srclen = strlen( df->data );
        *( buf + df->len ) = '\0';
        /*              */
        if ( srclen > df->len ) {               /*
*/
                memset( buf, '*', df->len );
                debug( ( stderr, "\t '%s': too
long for field (%d,%d)\n",
                                df->data, df->row,
df->col ) );
                return df->len;
        }

        memcpy( buf, df->data, srclen );
        memset( buf + srclen, ' ', df->len - srclen );
        /*          */

        return srclen;
}


/*                                      */

void insert_string( datafield_t *df, char *buf, int *len, int
ch, int *pos )
{
        int             end;
        int             i;

        if ( *len == df->len ) {                /*
*/
                return;
        }

        end = *pos;
        for ( i = *len - 1; i >= end; i-- ) {   /*
*/
                buf[i+1] = buf[i];
        }
        buf[end] = ch;
                /*          */
        ( *pos )++;
        /*              */
        ( *len )++;
        /*               */

        move( df->row, df->col + end );
        addnstr( buf + end, *len - end );
/*
        move( df->row, df->col );
        addstr( buf );
*/
        move( df->row, df->col + *pos );
}


/*                                      */
```

```
void delete_string( datafield_t *df, char *buf, int *len, int
*pos )
{
        int             end;
        int             i;

        if ( len == 0 ) {                       /*
*/
                return;
        }
        if ( *pos == *len ) {                   /*
*/
                return;
        /*                      */
        }

        end = *len - 1;
                /*              */
        for ( i = *pos; i < end; i++ ) {        /*
*/
                buf[i] = buf[i+1];
        }
        buf[end] = ' ';
                /*          */
        ( *len )--;
        /*                  */

        move( df->row, df->col );
        addstr( buf );
        move( df->row, df->col + *pos );
}


/*                                      */

void save_string( datafield_t *df, char *buf, int len )
{
        /* strcpy()
                                strncpy()
                              */

        memcpy( df->data, buf, len );
        *( df->data + len ) = '\0';
}


/*                      */

int edit_string_field( datafield_t *df )
{
        int             exitflag = FALSE;
        int             ch;
        int             len;
        /*                      */
        int             pos;
        /*                      */
        char            buf[128];

        len = make_string_field( buf, df );
        pos = len;

        move( df->row, df->col );
        attrset( ATTR_INPUT );
        addstr( buf );
        move( df->row, df->col + pos );
        refresh();

        while ( exitflag != TRUE ) {
                ch = getch();
                switch ( ch ) {
                case LF:
                case CR:
                case TAB:
                case KEY_UP:
                case CUU:
                case KEY_DOWN:
                case CUD:
        /*                  */
                        exitflag = TRUE;
                        break;
                case KEY_LEFT:
                case CUB:
        /*                  */
                        if ( pos > 0 ) {
                                                /*
*/
                                pos--;

                cursor_left1();
                        }
                        break;
                case KEY_RIGHT:
                case CUF:
        /*                  */
                        if ( ( pos < df->len
) && ( buf[pos] != ' ' ) ) {
                                                /*

*/

                pos++;

                cursor_right1();
                        }
                        break;
                case KEY_DC:
                case DEL:
                case EDEL:
        /*          */
                        delete_string( df,
buf, &len, &pos );
                        break;
                case BS:
        /*              */
                        /*
(cf. KEY_LEFT/KEY_DEL) */
                        if ( pos > 0 ) {
                                pos--;

                cursor_left1();

                delete_string( df, buf, &len, &pos );
                        }
                        break;
                default:
        /*          */
                        if ( isalnum( ch ) ) {

                insert_string( df, buf, &len, ch, &pos );
                        }
                        break;
                }
                refresh();
        }
        attrset( ATTR_BASE );
        save_string( df, buf, len );

        return ch;
}


/* display a set of fields */

void display_frame( int mode, const textfield_t *tf,
datafield_t *df )
{
        int             i = 0;
        /* tf  df           */
        char            buf[128];                       /*
*/

        debug1( ( stderr, "IN:\t display_frame(
mode: 0x%X, tf: %X, df: %X )\n",
                mode, tf, df ) );

        if ( mode & FR_TEXT ) {
                /*          text            */

                attrset( ATTR_BASE );
                for ( i = 0; !eofl( tf[i] ); i++ ) {
                        move( tf[i].row,
tf[i].col );
                        addstr( ( char *
)tf[i].data );

                        debug4( ( stderr,
"\t Text(%d): @(%2d,%2d) '%s'\n",
                                i,
tf[i].row, tf[i].col, tf[i].data ) );
                }
        }

        attrset( ATTR_DATA );
        if ( mode & FR_DATA ) {
                for ( i = 0; !eofl( df[i] ); i++ ) {
                        if ( !( df[i].flag &
FL_DATA ) ) {

                continue;
                        }

                        debug4( ( stderr,
                                                "\t
Data(%d): @(%2d,%2d), flag = %08X\n",
                                i,
df[i].row, df[i].col, df[i].flag ) );

                        if ( ( mode &
FR_UPDATE ) && ( !is_dirty( df[i] ) ) ) {

                continue;
                        }

                        if ( df[i].flag &
FL_STR ) {

                make_string_field( buf, &df[i] );
                        } else {

                make_number_field( buf, &df[i] );
                        }

                        move( df[i].row,
df[i].col );
                        addstr( buf );
```

```
                                clear_dirty( df[i] );
                        /* (--;;; */
                        }
                }

                attrset( ATTR_INPUT );
                if ( mode & FR_INPUT ) {
                        for ( i = 0; !eofl( df[i] ); i++ ) {
                                if ( !( df[i].flag &
FL_INPUT ) ) {

                continue;
                                }

                                debug4( ( stderr,
                                                "\t
Input(%d): @(%2d,%2d), flag = %08X\n",
                                                i,
df[i].row, df[i].col, df[i].flag ) );

                                if ( ( mode &
FR_UPDATE ) && ( !is_dirty( df[i] ) ) ) {

                continue;
                                }

                                if ( df[i].flag &
FL_STR ) {

                make_string_field( buf, &df[i] );
                                } else {

                make_number_field( buf, &df[i] );
                                }

                                move( df[i].row,
df[i].col );
                                addstr( buf );

                                clear_dirty( df[i] );
                        /* (--;;; */
                        }
                }

                attrset( ATTR_BASE );
                /* refresh(); */

                debug1( ( stderr, "OUT:\t
display_frame()\n" ) );
}


void query_frame( int start, datafield_t *df )
{
                int             exitflag = FALSE;
                int             pos = start;                    /* tf  df
*/
                int             i;
                int             rtn;
                char            buf[128];

                debug1( ( stderr, "IN:\t query_frame( start:
%d, df: %08X )\n",
                        start, df ) );

                while ( exitflag != TRUE ) {
                        if ( df[pos].flag & FL_STR ) {
                                rtn =
edit_string_field( &df[pos] );
                        } else if ( df[pos].flag &
FL_NUM ) {
                                rtn =
edit_number_field( &df[pos] );
                        }

                        debug4( ( stderr, "\t pos: %d,
rtn(key): %d\n", pos, rtn ) );
                        switch ( rtn ) {
                        case LF:
                        case CR:
                /*      */
                                exitflag = TRUE;
                                break;
                        case ESC:
                                break;
                        case KEY_UP:
                        case CUU:
                /*              */
                                i = pos - 1;
                                while ( i >= 0 && !(
df[i].flag & FL_INPUT ) ) {
                                        i--;
                                }
                                if ( i >= 0 ) {
                                        pos =
i;
                                }
                                break;
                        case TAB:
                        case KEY_DOWN:
                        case CUD:
                /*              */
                                i = pos + 1;
                                while ( !eofl( df[i] )
&& !( df[i].flag & FL_INPUT ) ) {
                                        i++;
                                }
                                if ( !eofl( df[i] ) ) {
                                        pos =
i;
                                }
                                break;
                        }
                }

                debug1( ( stderr, "OUT:\t query_frame()\n"
) );
}


/* Open curses and setup */

void init_screen()
{
                initscr();
                savetty();
                attrset( ATTR_BASE );
                cbreak();
                noecho();
                nonl();
#ifndef                 __linux__
                bkgd( ATTR_BASE );
                intrflush( stdscr, FALSE );
#endif
                keypad( stdscr, TRUE );
                nodelay( stdscr, FALSE );
                leaveok( stdscr, FALSE );
                refresh();
}


/* Clear the screen and close curses */

void close_screen()
{
                slk_clear();
                clear();
                refresh();
                resetty();
                endwin();
}

********** ui.h **********
/*
                ui.h : Header of field operation functions

                Version     1.00        1996/12/26 First
Edition
                ---------------------------
                Version     1.01        1997/05/20 for
term.c
                Version     1.10        1997/08/07 delete
color mode
                ---------------------------
                Version     2.00        1998/05/27
                Second Edition (for Tc6)
                Version     2.01        1998/06/05 delete
duplicate definitions
*/


#ifndef _UI_H_
#define _UI_H_

#ifdef TERM
                                /* for term.c */
# include "term.h"
#else           /* !TERM */
# ifdef __linux__
#  include <ncurses/curses.h>
# else
#  include <curses.h>
# endif
# define        EDEL        4

# define        CUU         16
# define        CUD         14
# define        CUF         6
# define        CUB         2
#endif          /* !TERM */


#define         SIMPLEVIEW

#define ATTR_BASE               A_NORMAL
#define ATTR_STATUS             A_NORMAL
#define ATTR_FRAME              A_NORMAL
#ifdef          SIMPLEVIEW
# define ATTR_DATA              A_NORMAL
# define ATTR_INPUT             A_UNDERLINE
#else
# define ATTR_DATA              A_UNDERLINE
# define ATTR_INPUT             A_STANDOUT
#endif
```

```
/* Frame option */

#define     FR_TEXT             0x00000001
            /* text fields (read only, const) */
#define     FR_DATA             0x00000002
            /* data fields (read only, var) */
#define     FR_INPUT  0x00000004            /*
input fields (writable, var) */
#define     FR_FULL             0x0000000f /* all
fields */
#define     FR_UPDATE           0x00000100
            /* updated fields only */
#define     FR_USERDEF          0x80000000
            /* user defined */


/* Field Categoly */

#define     FL_STR              (0x0001)    /*
alphabet & number */
#define     FL_SHORT (0x0002)   /* short number */
#define     FL_INT              (0x0004)    /*
int/long number */
#define     FL_DBL              (0x0008)    /*
double number */
#define     FL_NUM
            (FL_SHORT|FL_INT|FL_DBL)
#define     FL_SIGN             (0x0010)    /*
signed number */
#define     FL_BLANK (0x0020)   /* blank field */

#define     FL_TEXT             (0x0100)    /* text
(read only, const) */
#define     FL_DATA             (0x0200)    /* data
(read only, var) */
#define     FL_INPUT  (0x0400)  /* data (writable,
var) */
#define     FL_IGNORE           (0x0800)    /* data
(writable,  ignorable) */
#define     FL_DIRTY  (0x4000)  /* dirty field (for
FL_DATA) */
#define     FL_USRDEF           (0x8000)    /* user
defined */


/* macro definition */

#define     eofl( field_t )                 (
(field_t).row == -1 )
#define     clear_dirty( field_t )  ( (field_t).flag &=
~FL_DIRTY )
#define     set_dirty( field_t )    ( (field_t).flag |=
FL_DIRTY )
#define     is_dirty( field_t )     ( (field_t).flag &
FL_DIRTY )


/* structure definition */

typedef     struct {
            char                row;
            char                col;
            const char   *data;
} textfield_t;

typedef struct {
/*
            short               row;
            short               col;
            short               len;
            short               flen;
*/
            char                row;
            char                col;
            char                len;
            char                flen;
            unsigned    flag;
            char                *data;
} datafield_t;


/* function prototype */

void display_frame( int, const textfield_t *, datafield_t *
);
void query_frame( int, datafield_t * );
void init_screen();
void close_screen();

#endif
```

Appendix B: Server Source Code

```
******** 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;



******** 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;


******** makeTPCC *********
#
#               Makefile for test
#
#               Version       Beta2         1995/03/14
#
USR=/usr

#---TP=TP/base----------
ROOTDIR   = /opt/FSUNtpbs       # ROOTDIR =
/opt/uxptuxt
TPLIB = /opt/FSUNtpbs/lib

#---c language=fcc------------
FCCLIB=/opt/FSUNf90/lib
FCCBIN=/opt/FSUNf90/bin
CC = $(FCCBIN)/fcc
CCSLIB=$(FCCLIB)

#CCSLIB=$(USR)/ccs/lib
#CC           = cc
LIBS          = -lcurses
CCFLAGS                 = $(INCLUDEDIR) -s -O -K 3 -
K TMS -DDUR -DSOLARIS2

#----RDBMS=Symfo-------------
BASE=/opt/FSUNrdb2b
ICONVLIB=/opt/FSUNiconv/lib

SAMPLEPH=.
SRCPH=.
RDBLIB=$(BASE)/lib
RDBINC=$(BASE)/include

#----Cobol=Cobol85?-----------
COBLIB=$(USR)/lib
#COBLIB1=$(BASE)/uxpcbl85/lib
COBLIB2=/opt/uxpcblrt/lib
#LD_LIBRARY_PATH=$(RDBLIB):$(MEFTPLIB):$(TP
BASELIB)

#------------------------------
LD_LIBRARY_PATH=$(RDBLIB):$(FCCLIB):$(COBLI
B2):$(TPLIB):$(ICONVLIB)

MORE=-flrsu

all             : fldtbl.h TPCC

SVRFLAG  = "-l$(ROOTDIR)/include -L$(RDBLIB) -l
sqldrv -l sqldrv2"

BLDSVR = $(ROOTDIR)/bin/buildserver
MKFLDHDR = $(ROOTDIR)/bin/mkfldhdr

#TPCC:TPCC.o
#               $(BLDSVR) -o TPCC -f TPCC.o -f
OLINSERT.o -s TPCC -l$(SVRFLAG) -
l$(COBLIB2)#/libcobol.so -l$(COBLIB)/libdl.so

fldtbl.h : fldtbl
               $(MKFLDHDR) fldtbl

TPCC:TPCC_fml.o
```

```
		$(BLDSVR) -o TPCC -f TPCC_fml.o -f
OLINSERT.o -s TPCC -l$(SVRFLAG) -l/usr/lib/libc.so -
l$(COBLIB2)/libcobol.so -l$(COBLIB)/libdl.so

TPCC_fml.o:TPCC_fml.pc bench1.h bench2.h stored.h
		./sqlcc.fcc -W96 -I $(SRCPH) -t $(SRCPH)
-I $(SRCPH) -I $(ROOTDIR)/include TPCC_fml.pc -c
TPCC_fml.c -I $(ROOTDIR)/include -O -DUSE_FML -I
/opt/FSUNtpbs/include -D DP_SQLERR

##		sqlcc -I $(SRCPH) -t $(SRCPH) -I
$(SRCPH) -I $(ROOTDIR)/include -d $(RDBDB)
TPCC.pc -c -o TPCC.o -g -I /opt/uxptuxt/include


******** OLINSERT.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) BINARY.
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) BINARY.
001600     03 OL-D-ID       PIC  S9(04) BINARY.
001700     03 OL-W-ID       PIC  S9(04) BINARY.
001800     03 OL-NUMBER     PIC  S9(09) BINARY.
001500     03 OL-I-ID       PIC  S9(09) BINARY.
001600     03 OL-SUPPLY-W-ID PIC  S9(04) BINARY.
001700**   03  OL-DELIVERY-D  PIC  X(14)      .
001800     03 OL-QUANTITY   PIC  S9(04) BINARY.
001700**   03  DUMMY1       PIC  X(02)       .
001700     03 OL-AMOUNT     PIC  S9(09) BINARY.
001800     03 OL-DIST-INFO  PIC  X(24)       .
001700**   03  DUMMY2       PIC  X(03)       .
001000*
004100 01  O-OL-CNT       PIC  S9(04) BINARY.
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) BINARY.
001600     03 LIN-OL-D-ID       PIC  S9(04) BINARY.
001700     03 LIN-OL-W-ID       PIC  S9(04) BINARY.
001800     03 LIN-OL-NUMBER     PIC  S9(09)
BINARY.
001500     03 LIN-OL-I-ID       PIC  S9(09) BINARY.
001600     03 LIN-OL-SUPPLY-W-ID PIC  S9(04)
BINARY.
001700**   03  LIN-OL-DELIVERY-D  PIC  X(14)      .
001800     03 LIN-OL-QUANTITY   PIC  S9(04)
BINARY.
001700**   03  LIN-DUMMY1       PIC  X(02)       .
001700     03 LIN-OL-AMOUNT     PIC  S9(09)
BINARY.
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) BINARY.
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.
******** stored.h *********
/**********STORED
PROCEDURE*********************************************/
/** stored.h     COPYRIGHT FUJITSU LIMITED 1997
**/
/**    :                                          **/
/**    :                                          **/
/**    :        SymfoWARE RDB TPC-C Benchmark
**/
/**    :   SQL declare section for stored proceduer call
**/
/**    :        1996/09/06                        **/
/**             1997/03/13  Revision3.3 : Any
Error(Clause 2.3.6)      **/
/*****************************************************************
*****/

EXEC SQL BEGIN DECLARE SECTION;
char          state[6];
char          sqlmsg[257];
short         sqlmsg_ind;
int           errorpos;
short         errorpos_ind;          /* 1997.03.13    */

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             ;
```

```
******** 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)     **/
/*****************************************************************
*****/


#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"
#include <userlog.h>

#ifdef      USE_FML
                        /* 98.04.07 lch. */
# include   "fml.h"
# include   "fldtbl.h"
#endif

#include "bench3.h"                     /*    INSERT
960905 */
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

#define TIMES Gettimeofday(&tp); tv_st_sec=tp.tv_sec;
tv_st_usec=tp.tv_usec;

#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 ); \
    }
```

```
/*#define SOLARIS       */          /* 98.02.23  suzuki
*/

#ifdef UXP_DS  /* 98.02.23          suzuki */
#define Gettimeofday(a) gettimeofday(a)
#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     a*/

#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(){}

#if 1
                        /* 98.04.20 lch. */
#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);
```

```
#else
#define SLEEP_MIN       1000
#define SQLWAIT_O       usleep(    276 * SLEEP_MIN);
#define SQLWAIT_N       usleep(    326 * SLEEP_MIN);
#define SQLWAIT_N_C     usleep(      0 * SLEEP_MIN);
#define SQLWAIT_N_R     usleep(      0 * SLEEP_MIN);
#define SQLWAIT_P       usleep(    134 * SLEEP_MIN);
#define SQLWAIT_D       usleep(   1252 *
SLEEP_MIN);
#define SQLWAIT_S       usleep(    353 * SLEEP_MIN);
#endif

#endif

neworder_trans  *bpn;
payment_trans   *bpp;
orderstat_trans *bpo;
delivery_trans  *bpd;
stocklvl_trans  *bps;
#ifdef          USE_FML
                        /* 98.04.07 Ich. */
neworder_trans  nbuf;
payment_trans   pbuf;
orderstat_trans obuf;
delivery_trans  dbuf;
stocklvl_trans  sbuf;
#endif

struct timeval tp,tp_e;
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;

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 ;
}

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;

        /* 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,"/home/tpcc/delog/delivery_log%d",nu
m);
            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)
        {

            fprintf(delivery_handle,"%09d%03d
%09d%03d %d %d",
                  bpd->startsec,
                  bpd->startusec/1000,
                  tp_e.tv_sec,
                  tp_e.tv_usec/1000,
                  w_id,
                  o_carrier_id);

            for(d_id = 0; d_id < 10; d_id++)
            {
               fprintf(delivery_handle," %d
%d",d_id+1,result_o_id[d_id]);
            }
            fprintf(delivery_handle, "\n");
        }
        else
        {

            fprintf(delivery_handle,"%09d%03d
%09d%03d %d %d",
                  bpd->startsec,
                  bpd->startusec/1000,
                  0,
                  0,
                  w_id,
                  o_carrier_id);
            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("tpretum-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 ermo;
        FILE *handle;
        SQLSTATE[5] = 0 ;
        if (0 != strcmp(SQLSTATE,"00000") )
         {
           if (0 == strcmp(SQLSTATE,"40001")) /*
*/
            {
              return(1);
            }
           system("date >>/tmp/tpccerr");
           handle = fopen("/tmp/tpccerr","ab");
           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;
#ifdef  TRACE
        DP("tpsvrinit called pid=%d\n",getpid());
#endif
#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("CONNECTzzzzzzz(SQLSTATE) =
%s\n", SQLSTATE) ;

#ifdef          USE_SQL_MODE
                /* 98.02.23 suzuki */
        EXEC SQL COMMIT WORK ;
#endif
                  DP("COMMITyyyyy(SQLSTATE) =
%s\n", SQLSTATE) ;
        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);
        }
}

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;

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                    ;
         }
      }
      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_NORDER(: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. */
            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;
         }
       }

      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] ;
       if ( item_notfound == -1
        || item_notfound >  ol_number )
        {
         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)   ;
            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 ;
              }
            }
          }
        else
          {
                i_price[ol_number]        = 0             ;
                bpn->i_price[ol_number]    = 0            ;
                bpn->s_quantity[ol_number] = 0            ;
                ol_dist_info[0]           = '\0'          ;
          }

        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_info,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                        ;

        stmcpy(bpn->c_last,c_last,17)               ;
        stmcpy(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
      {
        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()                ;
  stmcpy(h_date, tc_s,14)       ;
```

```
        errorpos = 0            ;
#ifdef          USE_SQL_MODE
                /* 98.02.23 suzuki */
        EXEC SQL
          CALL TPCC_SCHEMA.Y_PAYMENT(: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, "199802121212 12" );
                strcpy( c_credit, "GC" );
#endif

        if ( memcmp(state,"00000",5) != 0 )
          {
            stmcpy(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        ;
    bpp->c_credit_lim = c_credit_lim   ;

    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 );
  } 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                      ;

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);
          bpo->ol_supply_w_id[ol_number]
                        = atoi(oljp-
>sqlvar[ol_number].ol_supply_w_id)   ;
          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       ;
        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                        ;

/* 98.06.08                                */
#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 )
     {
      stmcpy(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;
EXEC SQL COMMIT WORK;
#endif




    bps->errorpos    = 0 ;
    bps->sqlstate    = 0 ;
    /*EXEC SQL COMMIT WORK;*/
    return(1);

/* 98.06.08                    */
/* --SQLERR:NOT_FOUND */
ERR_S_DI:
    bps->errorpos = 203;
                    goto sqlerr;

ERR_S_STOL:
    bps->errorpos = 248;
                    goto sqlerr;
/* 98.06.08    */

not_found:
    DP("NOT FOUND IN STOCKLEVEL AT
%d\n",errorpos);
    fflush(stdout);
    bps->errorpos    = errorpos          ;
    bps->sqlstate    = atoi(SQLSTATE)    ;
    EXEC SQL ROLLBACK WORK;
    return(0);

sqlerr:
#ifdef DP_SQLERR
  DP("Stocklevel ERRPOS=%d
SQLSTATE=%s\n",errorpos,SQLSTATE);
#endif
    if(Error())
    {
        EXEC SQL ROLLBACK WORK;
        goto begin_tran;
    }
    bps->errorpos    = errorpos          ;
    bps->sqlstate    = atoi(SQLSTATE)    ;
    EXEC SQL ROLLBACK WORK;
    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));
    }
TPCC11(info)
TPSVCINFO    *info;
    {
    number = 11;
    return(TPCC(info,number));
    }
TPCC12(info)
TPSVCINFO    *info;
    {
    number = 12;
    return(TPCC(info,number));
    }
TPCC13(info)
TPSVCINFO    *info;
    {
    number = 13;
    return(TPCC(info,number));
    }
TPCC14(info)
TPSVCINFO    *info;
    {
    number = 14;
    return(TPCC(info,number));
    }
TPCC15(info)
TPSVCINFO    *info;
    {
    number = 15;
    return(TPCC(info,number));
    }
TPCC16(info)
TPSVCINFO    *info;
    {
    number = 16;
    return(TPCC(info,number));
    }
TPCC17(info)
TPSVCINFO    *info;
    {
    number = 17;
    return(TPCC(info,number));
    }
TPCC18(info)
TPSVCINFO    *info;
    {
    number = 18;
    return(TPCC(info,number));
    }
TPCC19(info)
TPSVCINFO    *info;
    {
    number = 19;
    return(TPCC(info,number));
    }
TPCC20(info)
TPSVCINFO    *info;
    {
    number = 20;
    return(TPCC(info,number));
    }
TPCC21(info)
TPSVCINFO    *info;
    {
    number = 21;
    return(TPCC(info,number));
    }
TPCC22(info)
TPSVCINFO    *info;
    {
    number = 22;
    return(TPCC(info,number));
    }
TPCC23(info)
TPSVCINFO    *info;
    {
    number = 23;
```

```
        return(TPCC(info,number));
        }
TPCC24(info)
TPSVCINFO       *info;
        {
        number = 24;
        return(TPCC(info,number));
        }
TPCC25(info)
TPSVCINFO       *info;
        {
        number = 25;
        return(TPCC(info,number));
        }
TPCC26(info)
TPSVCINFO       *info;
        {
        number = 26;
        return(TPCC(info,number));
        }
TPCC27(info)
TPSVCINFO       *info;
        {
        number = 27;
        return(TPCC(info,number));
        }
TPCC28(info)
TPSVCINFO       *info;
        {
        number = 28;
        return(TPCC(info,number));
        }
TPCC29(info)
TPSVCINFO       *info;
        {
        number = 29;
        return(TPCC(info,number));
        }
TPCC30(info)
TPSVCINFO       *info;
        {
        number = 30;
        return(TPCC(info,number));
        }
TPCC31(info)
TPSVCINFO       *info;
        {
        number = 31;
        return(TPCC(info,number));
        }
TPCC32(info)
TPSVCINFO       *info;
        {
        number = 32;
        return(TPCC(info,number));
        }
TPCC33(info)
TPSVCINFO       *info;
        {
        number = 33;
        return(TPCC(info,number));
        }
TPCC34(info)
TPSVCINFO       *info;
        {
        number = 34;
        return(TPCC(info,number));
        }
TPCC35(info)
TPSVCINFO       *info;
        {
        number = 35;
        return(TPCC(info,number));
        }
TPCC36(info)
TPSVCINFO       *info;
        {
        number = 36;
        return(TPCC(info,number));
        }
TPCC37(info)
TPSVCINFO       *info;
        {
        number = 37;
        return(TPCC(info,number));
        }
TPCC38(info)
TPSVCINFO       *info;
        {
        number = 38;
        return(TPCC(info,number));
        }
TPCC39(info)
TPSVCINFO       *info;
        {
        number = 39;
        return(TPCC(info,number));
        }
TPCC40(info)
TPSVCINFO       *info;
        {
        number = 40;
        return(TPCC(info,number));
        }



TPCCd1(info)
TPSVCINFO       *info;
        {
        number = 41;
        return(TPCC(info,number));
        }
TPCCd2(info)
TPSVCINFO       *info;
        {
        number = 42;
        return(TPCC(info,number));
        }
TPCCd3(info)
TPSVCINFO       *info;
        {
        number = 43;
        return(TPCC(info,number));
        }
TPCCs1(info)
TPSVCINFO       *info;
        {
        number = 44;
        return(TPCC(info,number));
        }
TPCCs2(info)
TPSVCINFO       *info;
        {
        number = 45;
        return(TPCC(info,number));
        }
TPCCs3(info)
TPSVCINFO       *info;
        {
        number = 46;
        return(TPCC(info,number));
        }
#endif

******** tra.wup.pc *********
/* TPCC  warming up ITEM 96.06.18 Kazutaka.Ozawa
*/
/* Function: READ ITEM TABLE. */

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("WUPI:@@
use 25 second\n");
                        item_s = we *
10000 - 10000 + 1 ;
                        item_e = we *
10000;
                        printf("for
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("WUPI:@@
use 90 second\n");
                        EXEC SQL
SELECT COUNT(*) INTO :i_c  FROM
TPCC_SCHEMA.ITEM;
                }
                EXEC SQL COMMIT WORK ;
                exit(0);
        }
}
******** Y_DELIVERY *********
-- /**********STORED
PROCEDURE*****************************************/
-- /** Y_DELIVERY   COPYRIGHT FUJITSU LIMITED
1997            **/
-- /**    :                                     **/
-- /**    :                                     **/
-- /**   I:    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)        |  |    g
P O
-- +---------------------------------------+  |
-- |      T v   X h ~ CHAR(1) "/"      |  |
-- +---------------------------------------+  ---
--  `                                `
-- +---------------------------------------+
--     j    o       p v   O       z X g    i u ` q b g ` q j
--       k k    i     D          j    K   O
--
*****************************************************************
**********
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  e [ u      A      l [ _
iNO_O_ID j
     --      R [ h i    A             l [ _ E   R [ h j
                          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 e [ u      A@NO_O_ID     v
R [ h
     --   i             l [ _    R [ h j      i P  j
                          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 e [ u      AOL_AMOUNT     v
|
                          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 e [ u     Y     R [ h   X V
                          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  e [ u      A@NO_O_ID     v      R [
h
     --     i P   j A       o
                          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  e [ u     Y     R [ h   X V
                          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)  z          l [ _    s    Customer e [ u   E   R [ h
X V
                          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;

******** Y_NORDER *********
-- /**********STORED
PROCEDURE***********************************************/
-- /** Y_NORDER     COPYRIGHT FUJITSU LIMITED
1997                 **/
-- /**    :                                          **/
-- /**    :                                          **/
-- /**    I:      SymfoWARE RDB TPC-C Benchmark
**/
-- /**  @ \:    NewOrder                             **/
-- /**    :     1996/10/12                           **/
-- /**          1997/03/13  Revision3.3 : Any
Error(Clause 2.3.6)     **/
--
/*****************************************************************
*****/


--
*****************************************************************
*********
-- #S_JOIN                 VARCHAR(1215)
-- +---------------------------------------+
-- | sqllen       short              |
-- +-sqlver-------------------------------+ ---
-- | S_QUANTITYn     CHAR(6)          | |
-- +---------------------------------------+ |
-- | S_DISTn         CHAR(24)         | |    g         P T
-- +---------------------------------------+ |
-- | S_DATAn         CHAR(50)         | |
-- +---------------------------------------+ |
-- |       T v   X h ~CHAR(1) "/"       | |
-- +---------------------------------------+ ---
-- |                                       |
-- `                                       `
-- +---------------------------------------+
--    j   o      p v  O      z X g    i u ` q b g ` q j
--      k k    i     D         j    K   O
--
-- #I_JOIN               VARCHAR(1215)
-- +---------------------------------------+
-- | sqllen       short              |
-- +-sqlver-------------------------------+ ---
-- | I_PRICEHn       CHAR(6)          | |
-- +---------------------------------------+ |
-- | I_NAMEn         CHAR(24)         | |    g         P
T
-- +---------------------------------------+ |
-- | I_DATAn         CHAR(50)         | |
-- +---------------------------------------+ |
-- |       T v   X h ~CHAR(1) "/"       | |
-- +---------------------------------------+ ---
-- |                                       |
-- `                                       `
-- +---------------------------------------+
--    j   o      p v  O      z X g    i u ` q b g ` q j
--      k k    i     D         j    K   O
--
*****************************************************************
**********
EXEC SQL
CREATE PROCEDURE
TPCC_SCHEMA.Y_NORDER(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    ;

-- (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_DIST_04,S_DI
ST_05,

S_DIST_06,S_DIST_07,S_DIST_08,S_DIST_09,S_DI
ST_10,

S_YTD,S_ORDER_CNT,S_REMOTE_CNT,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
          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 = @MATCH_TBL_CNT ;
    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_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;
                        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  ;

-- Home Warehouse PROCESS END

   IF  #R_CNT = 0 THEN
    GOTO DISTRICT_PROC ;
   END IF;

REMORT_PROC:
-- Remote Warehouse process start
-- ( [              O       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 R2     ;
     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_DI
ST_05,

S_DIST_06,S_DIST_07,S_DIST_08,S_DIST_09,S_DI
ST_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 ;

--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;
******** Y_ODERSTAT *********
-- /**********STORED
PROCEDURE*****************************************/
-- /** Y_ODERSTAT   COPYRIGHT FUJITSU LIMITED
1997                 **/
-- /**    :                                      **/
-- /**    :                                      **/
-- /**    I:      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)
--  +-----------------------------------+
--  | sqlen       short                 |
--  +-sqlver-----------------------------+  ---
--  | #OL_I_IDn       CHAR(7)           |  |
--  +-----------------------------------+  |
--  | #OL_AMOUNTn     CHAR(8)           |  |
--  +-----------------------------------+  |    g         P T
--  | #OL_SUPPLY_W_IDn CHAR(4)          |  |
--  +-----------------------------------+  |
--  | #OL_QUANTITYn    CHAR(4)          |  |
--  +-----------------------------------+  |
--  | #OL_DELIVERYn    CHAR(14)         |  |
--  +-----------------------------------+  |
--  |      T v   X h ~CHAR(1) "/"       |  |
--  +-----------------------------------+  ---
--  `                                   `
--  +-----------------------------------+
--     j    o       p v   O       z X g     i u ` q b g ` q j
--       k k     i      D          j     K   O
--
*********************************************************************
**********
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;
******** Y_PAYMENT *********
-- /**********STORED
PROCEDURE*****************************************/
-- /** Y_PAYMENT    COPYRIGHT FUJITSU LIMITED
1997              **/
-- /**    :                                      **/
-- /**    :                                      **/
```

```
-- /**   I:     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(12,2),
                    OUT  #C_DISCOUNT
SMALLINT,
                    INOUT #C_BALANCE
DECIMAL(12,2),
                    INOUT #C_YTD_PAYMENT
DECIMAL(12,2),
                    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(12,2);
    DECLARE @D_YTD          DECIMAL(12,2);
    DECLARE @C_DATA476      CHAR(476);
    DECLARE @H_DATA         CHAR(24);

-- CUSTOMER  e [ u   J [ \
    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  e [ u   i    j   A
      -- Customer Last Name    v     R [ h  C_FIRST
A
      -- NAMECOUNT/ Q       R [ h       o
      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;

      -- Customer        z i#C_BALANCE j  X V
      -- Customer       z     l i#C_YTD_PAYMENT j  X
V
      -- Customer            l i#C_PAYMENT_CNT j  X
V
      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_DATA476
             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 V   c_data          .
        SET   #C_DATA   = CAST(#C_ID    AS
CHAR(5))
                    || CAST(#C_D_ID   AS CHAR(2))
                    || CAST(#C_W_ID   AS CHAR(3))
                    || CAST(#D_ID     AS CHAR(2))
                    || CAST(#W_ID     AS CHAR(3))
                    || CAST(#H_AMOUNT AS CHAR(7))
                    || ' '
                    || @C_DATA476;

        -- (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  e [u    X V i    z    l t B [   h j
      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  e [u    J [ \
                        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  e [u    X V i    z    l t B [   h j
      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  e [u    }       R [ h i#H_DATA j
      --    HISTORY  e [u    P   R [ h }
      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;
******** Y_STOCKLV *********
-- /**********STORED
PROCEDURE*********************************************/
-- /** Y_STOCKLV    COPYRIGHT FUJITSU LIMITED
1997                **/
-- /**    :                                        **/
-- /**    :                                        **/
-- /**    I:     SymfoWARE RDB TPC-C Benchmark
**/
-- /**  @ \:   StockLevel                          **/
-- /**    :     1996/10/12                         **/
-- /**          1997/03/13  Revision3.3 : Any
Error(Clause 2.3.6)     **/
--
/*****************************************************************
*****/

EXEC SQL
CREATE PROCEDURE
TPCC_SCHEMA.Y_STOCKLV(OUT   #STATE
CHAR(5),
                        INOUT #ERRPOS
INTEGER,
                        IN   #W_ID          SMALLINT,
                        IN   #D_ID          SMALLINT,
                        IN   #THRESHOLD
INTEGER,
                        INOUT #LOW_STOCK
INTEGER )

STOCK_LEVEL:BEGIN
-- DECLARE
      DECLARE SQLSTATE        CHAR(5)
DEFAULT '00000';
      DECLARE @O_ID           INTEGER;
      DECLARE @TMP_O_ID       INTEGER;
      DECLARE @T02        INTEGER;
      DECLARE @T03        INTEGER;
      DECLARE @T04        INTEGER;
      DECLARE @T05        INTEGER;
      DECLARE @T06        INTEGER;
      DECLARE @T07        INTEGER;
      DECLARE @T08        INTEGER;
      DECLARE @T09        INTEGER;
      DECLARE @T10        INTEGER;
      DECLARE @T11        INTEGER;
      DECLARE @T12        INTEGER;
      DECLARE @T13        INTEGER;
      DECLARE @T14        INTEGER;
      DECLARE @T15        INTEGER;
      DECLARE @T16        INTEGER;
      DECLARE @T17        INTEGER;
      DECLARE @T18        INTEGER;
      DECLARE @T19        INTEGER;


-- (1) DISTRICT   teble select
                        WHENEVER SQLERROR
GOTO ERR_S_DI;
                        WHENEVER NOT FOUND
GOTO ERR_S_DI;
      SELECT D_NEXT_O_ID
          INTO @O_ID
          FROM TPCC_SCHEMA.DISTRICT
          WHERE D_W_ID = #W_ID
            AND D_ID   = #D_ID;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
-- (2) ORDERLINE  teble select
-- (3) STOCK      teble select and count ITEM

      SET   #LOW_STOCK = 0;
      SET   @TMP_O_ID  = @O_ID - 20;
      SET   @O_ID      = @O_ID - 1 ;
      SET   @T19     = @O_ID - 1 ;
      SET   @T18     = @T19 - 1 ;
      SET   @T17     = @T18 - 1 ;
      SET   @T16     = @T17 - 1 ;
      SET   @T15     = @T16 - 1 ;
      SET   @T14     = @T15 - 1 ;
      SET   @T13     = @T14 - 1 ;
      SET   @T12     = @T13 - 1 ;
      SET   @T11     = @T12 - 1 ;
      SET   @T10     = @T11 - 1 ;
      SET   @T09     = @T10 - 1 ;
      SET   @T08     = @T09 - 1 ;
      SET   @T07     = @T08 - 1 ;
      SET   @T06     = @T07 - 1 ;
      SET   @T05     = @T06 - 1 ;
      SET   @T04     = @T05 - 1 ;
      SET   @T03     = @T04 - 1 ;
      SET   @T02     = @T03 - 1 ;

                        WHENEVER SQLERROR
GOTO ERR_S_STOL;
                        WHENEVER NOT FOUND
GOTO ERR_S_STOL;
      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    = #W_ID
            AND S_QUANTITY < #THRESHOLD ;
                        WHENEVER SQLERROR
CONTINUE;
                        WHENEVER NOT FOUND
CONTINUE;
      COMMIT WORK         ;
      SET #STATE = '00000' ;
      LEAVE STOCK_LEVEL    ;

--SQLERR:NOT_FOUND
ERR_S_DI:
      SET #ERRPOS = 203    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;
      LEAVE STOCK_LEVEL    ;

ERR_S_STOL:
      SET #ERRPOS = 248    ;
      SET #STATE  = SQLSTATE;
      ROLLBACK WORK        ;

END STOCK_LEVEL
END-EXEC;
******** Y_stored.sh *********
 set -x
rdbddlex -d TPCC  -x Y_NORDER
rdbddlex -d TPCC  -x Y_PAYMENT
```

```
rdbddlex -d TPCC  -x Y_ODERSTAT
rdbddlex -d TPCC  -x Y_DELIVERY
rdbddlex -d TPCC  -x Y_STOCKLV
```

Appendix C: RTE Scripts

```
#
#  tpcC.conf : configration file for TPC-C
#
#

STARTGROUP = sync , 1
        STARTRTE
                RTEHOST   = rte04
                STARTSUT
                        SUTHOST
= cl04a,500
                        SUTLOGIN
= tpcc

        SUTPASSWD = tpcc
                        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,9237
#WAREHOUSE SCALE
        VAL         = U1I = 800
#RAMP-UP TIME
        VAL         = U2I = 0
#MEASUERMENT TIME
        VAL         = U3I = 5400
#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

```
******** conbf.sh ********
:
echo "# WORK FILE for rdbconbf" >conbf_wk1
echo "# WORK FILE for rdbconbf" >conbf_wk2
echo "# WORK FILE for rdbconbf" >conbf_wk3
echo "# WORK FILE for rdbconbf" >conbf_wk4

mkconbf2 WAREHOUSE   28 W    7 1
>>conbf_wk1
mkconbf2 DISTRICT    28 D    7 1 >>conbf_wk1
mkconbf2 CUSTOMER    168 C    7 6
>>conbf_wk1
mkconbf2 CUSTOMER_X 168 C_IX 7 6
>>conbf_wk1
mkconbf2 ORDERS      168 O    7 6
>>conbf_wk1
mkconbf2 ORDERS_IX   168 O_IX 7 6
>>conbf_wk1
mkconbf2 NEWORDER    168 NO   7 6
>>conbf_wk1
mkconbf2 NEWORDER_X 168 NO_IX 7 6
>>conbf_wk1
mkconbf2 STOCK       56 S    14 2
>>conbf_wk1
mkconbf2 ITEM        1 I     1 1 >>conbf_wk1

mkconbf2 ORDERLIN    168 OL   14 6
>>conbf_wk1
##mkconbf2 ORDERLIN_X 168 OL_IX 14 6
>>conbf_wk1
mkconbf2 HISTORY     168 H    7 6
>>conbf_wk1

timex rdbconbf -f conbf_wk1
#timex rdbconbf -f conbf_wk2
#timex rdbconbf -f conbf_wk3
#timex rdbconbf -f conbf_wk4


******** crbuf.aa ********
W_1         1K          240         100
            100         1 2147483646 fixed
W_2         1K          240         100
            100         1 2147483646 fixed
W_3         1K          240         100
            100         1 2147483646 fixed
W_4         1K          240         100
            100         1 2147483646 fixed
W_5         1K          240         100
            100         1 2147483646 fixed
W_6         1K          240         100
            100         1 2147483646 fixed
W_7         1K          240         100
            100         1 2147483646 fixed

D_1         1K          2400        100
            100         1 2147483646 fixed
D_2         1K          2400        100
            100         1 2147483646 fixed
D_3         1K          2400        100
            100         1 2147483646 fixed
D_4         1K          2400        100
            100         1 2147483646 fixed
D_5         1K          2400        100
            100         1 2147483646 fixed
D_6         1K          2400        100
            100         1 2147483646 fixed
D_7         1K          2400        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_IX_1      32K         24          100
            100         1 2147483646 fixed
C_IX_2      32K         24          100
            100         1 2147483646 fixed
C_IX_3      32K         24          100
            100         1 2147483646 fixed
C_IX_4      32K         24          100
            100         1 2147483646 fixed
C_IX_5      32K         24          100
            100         1 2147483646 fixed
C_IX_6      32K         24          100
            100         1 2147483646 fixed
C_IX_7      32K         24          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_IX_1      32K         144         100 100
            1 2147483646 fixed
O_IX_2      32K         144         100 100
            1 2147483646 fixed
O_IX_3      32K         144         100 100
            1 2147483646 fixed
O_IX_4      32K         144         100 100
            1 2147483646 fixed
O_IX_5      32K         144         100 100
            1 2147483646 fixed
O_IX_6      32K         144         100 100
            1 2147483646 fixed
O_IX_7      32K         144         100 100
            1 2147483646 fixed

NO_1        1K          3           90
            80          1 2147483646 fixed
NO_2        1K          3           90
            80          1 2147483646 fixed
NO_3        1K          3           90
            80          1 2147483646 fixed
NO_4        1K          3           90
            80          1 2147483646 fixed
NO_5        1K          3           90
            80          1 2147483646 fixed
NO_6        1K          3           90
            80          1 2147483646 fixed
NO_7        1K          3           90
            80          1 2147483646 fixed

NO_IX_1     32K         28          100
            100         1 2147483646 fixed
NO_IX_2     32K         28          100
            100         1 2147483646 fixed
NO_IX_3     32K         28          100
            100         1 2147483646 fixed
NO_IX_4     32K         28          100
            100         1 2147483646 fixed
NO_IX_5     32K         28          100
            100         1 2147483646 fixed
NO_IX_6     32K         28          100
            100         1 2147483646 fixed
NO_IX_7     32K         28          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
S_13        1K          3           90
            80          1 2147483646 fixed
S_14        1K          3           90
            80          1 2147483646 fixed

I_1         1K          14300       100
            100         1 2147483646 fixed
******** fssqlenv ********
:
; All Rights Reserved, Copyright(c) FUJITSU
1993, 1994, 1995
```

```
; All Rights Reserved, Copyright(c) PFU     1993,
1994, 1995
;
;     :
;
;
;           :           ';'
;           :
;           : 1                       1024
;
MAX_CONNECT_TCP=(290)
MAX_CONNECT_SYS=(350)
;MAX_CONNECT_SYS=(256)
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)
********  mkconbf2.c ********
#include <stdio.h>
#include <ermo.h>

main(int argc, char *argv[] )
{
            int i,j, dsi_n, pool_n;
            char *dsi_p, *pool_p;
            int cnt_dsi, cnt_wh, dsi_ren;

            if( argc != 6 ){
                        fprintf(stderr,
      "usage:       DSI      DSI                    "
                                                          "
DSI >       \n");
                        exit(1);
            }
            dsi_p = argv[1];
            dsi_n = atoi(argv[2]);
            pool_p = argv[3];
            pool_n = atoi(argv[4]);
            dsi_ren = atoi(argv[5]);

            for( i=1,j=1,cnt_wh=1; i<=dsi_n;
i++,cnt_wh++){
                        if( cnt_wh> dsi_ren ){
                                    cnt_wh=1;
                                    j++;
                                    if( j> pool_n
){

            j=1;
                                    }
                        }


            printf("TPCC.%s_%d_DSI %s_%d
\n",dsi_p,i, pool_p, j);
            }
            return(0);
}
********  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
********  rdbbuf.RCV ********
#
# All Rights Reserved, Copyright(c) FUJITSU
1993, 1994, 1995
# All Rights Reserved, Copyright(c) PFU     1993,
1994, 1995
#
#     :
#
#
#           : '#'
#           :
#           : 1                       1024
#
#    <<       >>
#                                 =
#
#######################################
#######################################nrk
BUFFER1K  = 4096
BUFFER2K  = 1024
BUFFER4K  = 20480
BUFFER8K  = 10240
BUFFER16K = 2048
BUFFER32K = 4096
********  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_ESUB    = 351232 ,1024  ,1024000000 #60
BCM_GPCT    = 4096   ,4096  ,1024000000
#508
BCM_IOPROC  = 71680  ,4096  ,1024000000
#380
BCM_LOGAREA = 0      ,1     ,1024000000
#1020
BCM_LOGLIST = 0      ,1     ,1024000000 #252
BCM_PGC     = 176128 ,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_FGRP    = 43008  ,4096  ,1024000000
#252
CCR_IDT     = 849920 ,4096  ,1024000000
#912(0x390)
CCR_KAIOCB  = 184320 ,1024  ,1024000000
#92
CCR_LWPIDT  = 157696 ,4096  ,1024000000
#112(0x70)
CCR_POLMCTL = 4184064 ,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_WLIST   = 17408  ,1024  ,1024000000
#28(0x1c)
CCR_WPID    = 276480 ,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_DSIL    = 2668544 ,1024  ,1024000000
#60
DSM_DSVP    = 7168   ,1024  ,1024000000 #60
DSM_DSVQ    = 50176  ,1024  ,1024000000 #92
DSM_DSWH    = 0      ,4096  ,1024000000 #2044
DSM_DSWP    = 0      ,1024  ,1024000000 #124
DSM_DUSI    = 12288  ,1024  ,1024000000 #60
DSM_DWFL    = 0      ,1024  ,1024000000 #60
DSM_DWUI    = 0      ,1024  ,1024000000 #60
LCM_LOGCNTL = 47104  ,4096  ,1024000000
#252
SCI_CMD     = 73728  ,4096  ,1024000000 #508
SCI_CONBF   = 4096   ,4096  ,1024000000 #508
SSV_IINF    = 20480  ,1024  ,1024000000 #124
TCM_TRAN    = 43008  ,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_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_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

******** RDBS.rcv ********
#!/usr/bin/sh -xf
RDBDB=TPCC
sync
rehash

RDB_SDP_MEM_SIZE_K=5200
export RDB_SDP_MEM_SIZE_K

#MBSPERKEY=6500
MBSPERKEY=5500
#MBSPERKEY=7000 #NG
export MBSPERKEY

MBSFIXTIME=220000
#MBSFIXTIME=300000 #NG?
#MBSFIXTIME=150000
export MBSFIXTIME

#RDBMBSFIXCOPYNOP=1
#export RDBMBSFIXCOPYNOP

#RCV_LOGBUFNUM=512       #for down rcv
RCV_LOGBUFNUM=1024       #for down rcv

###
export RCV_LOGBUFNUM

##
RCV_MULTINUM=32
#RCV_MULTINUM=64
#RCV_MULTINUM=128

#
#
export RCV_MULTINUM


RDBBUFPATH=/rdbptc/tpcc80/tpcc/b-
onsrc/rdbcwbuf.wk
export RDBBUFPATH

#R_LOCK_TBL=100,2937    #sql.env -
>DSO_LOCK to renkei
#export R_LOCK_TBL

#   yield              97.10.06
SEINO_YIELD=1
export SEINO_YIELD

#RDBSHMMMU_NOP=1
#export RDBSHMMMU_NOP

#              16MB

RDB_LOCAL_LP_ADDR=0x4000000
export RDB_LOCAL_LP_ADDR

##RDB_LOCAL_LP_SIZE=992
RDB_LOCAL_LP_SIZE=794   #980601

export RDB_LOCAL_LP_SIZE

#RDB_LOCAL_LP_ADDR2=0x42000000
RDB_LOCAL_LP_ADDR2=0x36000000
#980601
export RDB_LOCAL_LP_ADDR2
###RDB_LOCAL_LP_SIZE2=656
RDB_LOCAL_LP_SIZE2=528    #980614
export RDB_LOCAL_LP_SIZE2

#
RDBMBSBPPATH=/rdbptc/tpcc80/tpcc/b-
onsrc/mbs_tmp

RDBMBSSPACENUM=7
RDBMBSSDPNUM=7
###RDBMBSCTLADDR=0XEC000000
###RDBMBSCTLADDR=0XE7000000
RDBMBSCTLADDR=0XEB000000
#RDBMBSBUFSIZE=536870912
#RDBMBSBUFSIZE=268435456
RDBMBSBUFSIZE=409600000   #390MB for
49000 * 4k * 14pool
##RDBMBSBUFSIZE=471859200  #450MB

#RDBMBSBUFADDR=0x42000000
RDBMBSBUFADDR=0x36000000 #980601

RDBMBSBPCXNUM=20
RDBMBSBUFNUM=1
RDBMBSMAINCTLADDR=0X4000000
RDBMBSMAINCTLSHMKEY=0XEEEF9876
RDBMBSWBOXNUM=10000
RDBFIXBUFSHMKEY=0xdb000010

###RDBFIXBUFSHMADDR=0x5f000000  ##
98.06.01 for 2240MB
RDBFIXBUFSHMADDR=0x57000000  ##
98.06.14 for 2368MB

RDBFIXBUFSHMKEY2=0xdb000011  ## for
2048M <= FIXBUFUFMEMSIZE

###RDBFIXBUFSHMADDR2=0xa5000000 ##
98.06.01 for 2240MB
RDBFIXBUFSHMADDR2=0xa1000000 ##
98.06.14 for 2368MB

export RDBMBSBPPATH
export RDBMBSSPACENUM
export RDBMBSSDPNUM
export RDBMBSCTLADDR
export RDBMBSBUFSIZE
export RDBMBSBUFADDR
export RDBMBSBPCXNUM
export RDBMBSBUFNUM
export RDBMBSMAINCTLADDR
```

```
export RDBMBSMAINCTLSHMKEY
export RDBMBSWBOXNUM
export RDBFIXBUFSHMKEY
export RDBFIXBUFSHMADDR
export RDBFIXBUFSHMKEY2
export RDBFIXBUFSHMADDR2

date; timex rdbstart
date; #-----------------------------------------------
******** RDBSTART ********
#!/usr/bin/sh -xf
RDBDB=TPCC
sync

RDBFOPENMAX=230
export RDBFOPENMAX
KARI_CL_ST=570
export KARI_CL_ST

#RDB_SDP_MEM_SIZE_K=10000
#RDB_SDP_MEM_SIZE_K=2000
#export RDB_SDP_MEM_SIZE_K

#MBSPERKEY=6500
MBSPERKEY=5500  #KANSA BASE
#MBSPERKEY=7000 #NG
export MBSPERKEY

MBSFIXTIME=220000 #KANSA BASE
#MBSFIXTIME=990000
#MBSFIXTIME=300000 #NG?
#MBSFIXTIME=150000
export MBSFIXTIME

#RDBMBSFIXCOPYNOP=1
#export RDBMBSFIXCOPYNOP

###RCV_LOGBUFNUM=512        #for down rcv
###export RCV_LOGBUFNUM

##RCV_MULTINUM=64
##export RCV_MULTINUM


RDBBUFPATH=/rdbptc/tpcc80/tpcc/b-
onsrc/rdbcwbuf.wk
export RDBBUFPATH

#R_LOCK_TBL=100,2937     #sql.env -
>DSO_LOCK to renkei
#export R_LOCK_TBL

#    yield              97.10.06
SEINO_YIELD=1
export SEINO_YIELD

#RDBSHMMMU_NOP=1
#export RDBSHMMMU_NOP

#              16MB

RDB_LOCAL_LP_ADDR=0x4000000
export RDB_LOCAL_LP_ADDR

##RDB_LOCAL_LP_SIZE=992
RDB_LOCAL_LP_SIZE=794    #980601
export RDB_LOCAL_LP_SIZE

#RDB_LOCAL_LP_ADDR2=0x42000000
RDB_LOCAL_LP_ADDR2=0x36000000
#980601
export RDB_LOCAL_LP_ADDR2

###RDB_LOCAL_LP_SIZE2=656      #980601
RDB_LOCAL_LP_SIZE2=528     #980614

export RDB_LOCAL_LP_SIZE2

#
RDBMBSBPPATH=/rdbptc/tpcc80/tpcc/b-
onsrc/mbs_tmp

RDBMBSSPACENUM=7
RDBMBSSDPNUM=7
###RDBMBSCTLADDR=0XEC000000
###RDBMBSCTLADDR=0XE7000000
RDBMBSCTLADDR=0XEB000000
#RDBMBSBUFSIZE=536870912
#RDBMBSBUFSIZE=268435456
RDBMBSBUFSIZE=409600000    #390MB for
49000 * 4k * 14pool
##RDBMBSBUFSIZE=471859200  #450MB

#RDBMBSBUFADDR=0x42000000
RDBMBSBUFADDR=0x36000000 #980601

RDBMBSBPCXNUM=20
RDBMBSBUFNUM=1
RDBMBSMAINCTLADDR=0X4000000
RDBMBSMAINCTLSHMKEY=0XEEEF9876
RDBMBSWBOXNUM=1000
RDBFIXBUFSHMKEY=0xdb000010

###RDBFIXBUFSHMADDR=0x5f000000  ##
98.06.01 for 2240MB
RDBFIXBUFSHMADDR=0x57000000  ##
98.06.14 for 2368MB

RDBFIXBUFSHMKEY2=0xdb000011  ## for
2048M <= FIXBUFUFMEMSIZE

###RDBFIXBUFSHMADDR2=0xa5000000 ##
98.06.01 for 2240MB
RDBFIXBUFSHMADDR2=0xa1000000 ##
98.06.14 for 2368MB

export RDBMBSBPPATH
export RDBMBSSPACENUM
export RDBMBSSDPNUM
export RDBMBSCTLADDR
export RDBMBSBUFSIZE
export RDBMBSBUFADDR
export RDBMBSBPCXNUM
export RDBMBSBUFNUM
export RDBMBSMAINCTLADDR
export RDBMBSMAINCTLSHMKEY
export RDBMBSWBOXNUM
export RDBFIXBUFSHMKEY
export RDBFIXBUFSHMADDR
export RDBFIXBUFSHMKEY2
export RDBFIXBUFSHMADDR2

date; timex rdbstart
date; #-----------------------------------------------

rdbcrbufcw cw_env.comp

date


#csh sh.crbuf.mk
#csh sh.crbuf.mk.MBS.1680WH
#csh sh.crbuf.mk.MBS.1680WH.2
#csh sh.crbuf.mk.MBS.1680WH.3
#csh sh.crbuf.mk.MBS.1680WH2
#csh sh.crbuf.mk.MBS.1680WH2.2
#csh sh.crbuf.mk.MBS.OL.L_IX.O
#csh sh.crbuf.mk.MBS.OL.L_IX.O.Smain
##csh sh.crbuf.mk.MBS.OL.L_IX.Smain.4K
#csh sh.crbuf.mk.MBS.OL.L_IX.Smain.Sp
######csh sh.crbuf.mk.MBS.Smain.4K
#csh sh.crbuf.mk.MBS.Smain.4K_OLmain
#csh sh.crbuf.mk.MBS.Smain.4K_OLmain2
#csh sh.crbuf.mk.MBS.Smain.4K_OLmain3
#csh sh.crbuf.mk.MBS.Smain.4K_OLx  #Notest
####csh sh.crbuf.mk.MBS.Smain.4K_OLx2
#csh sh.crbuf.mk.MBS.Smain.4K_half2
#csh sh.crbuf.mk.noMBS

########################################
###### for PS
#csh sh.crbuf.mk.PS.1
#csh sh.crbuf.mk.PS.2.o_x
###csh sh.crbuf.mk.PS.2.Smain
#csh sh.crbuf.mk.PS.2.Smain.CaB
#csh sh.crbuf.mk.PS.2.Smain.OL32k
#csh sh.crbuf.mk.PS.2.Smain.OL32k.2
#csh sh.crbuf.mk.PS.2.Smain.OL32k.3.8h
#csh sh.crbuf.mk.980702            #ichibu 7
bunkatsu
csh sh.crbuf.mk.980714            #ichibu 7
bunkatsu

sh sh.crbuf


timex rdbcrbf -f crbuf.aa  2>crbf.wk.out1  #WH
grep -v qdg02630 crbf.wk.out1

sh conbf.sh
#sh conbf.sh.MBS

date
sar -r 1 1
swap -s

#ITEM
RDBMSG=E
export RDBMSG
wupi.sh #(          ITEM READ)

sar -r 1 1
swap -s

crbfadd.sh

cwenvchk

date

echo "RDBSTART env" >
res.doc/RDBSTART.env
env >> res.doc/RDBSTART.env
cwenvchk >>res.doc/RDBSTART.env

rdblog -V -a > res.doc/arclog_s
******** RDBSTART.ddl ********
:
```

```
set -x
date

#               16MB
#RDBSHMMMU_NOP=1
#export RDBSHMMMU_NOP

RDB_SDP_MEM_SIZE_K=10000
export RDB_SDP_MEM_SIZE_K

RDB_LOCAL_LP_ADDR=0x4000000
export RDB_LOCAL_LP_ADDR

RDB_LOCAL_LP_SIZE=794   #980601

export RDB_LOCAL_LP_SIZE

##RDB_LOCAL_LP_ADDR2=0x36000000
#980601
##export RDB_LOCAL_LP_ADDR2
##RDB_LOCAL_LP_SIZE2=656
##export RDB_LOCAL_LP_SIZE2

#
RDBMBSBPPATH=/rdbptc/tpcc80/tpcc/b-
onsrc/mbs_tmp

##RDBMBSSPACENUM=7
##RDBMBSSDPNUM=7
RDBMBSSPACENUM=1   #for Load
RDBMBSSDPNUM=8     #for Load

##RDBMBSCTLADDR=0XEB000000
RDBMBSCTLADDR=0X72000000  #98.06.03 for
Load

##RDBMBSBUFSIZE=409600000   #390MB for
49000 * 4k * 14pool
RDBMBSBUFSIZE=4096000   #98.06.03 for
Load

RDBMBSBUFADDR=0x36000000 #980601

RDBMBSBPCXNUM=20
RDBMBSBUFNUM=1
RDBMBSMAINCTLADDR=0X4000000
RDBMBSMAINCTLSHMKEY=0XEEEF9876
RDBMBSWBOXNUM=1000
RDBFIXBUFSHMKEY=0xdb000010
RDBFIXBUFSHMADDR=0x5f000000  ##
98.06.01 for 2240MB

##RDBFIXBUFSHMKEY2=0xdb000011  ## for
2048M <= FIXBUFUFMEMSIZE
##RDBFIXBUFSHMADDR2=0xa5000000 ##
98.06.01 for 2240MB

export RDBMBSBPPATH
export RDBMBSSPACENUM
export RDBMBSSDPNUM
export RDBMBSCTLADDR
export RDBMBSBUFSIZE
export RDBMBSBUFADDR
export RDBMBSBPCXNUM
export RDBMBSBUFNUM
export RDBMBSMAINCTLADDR
export RDBMBSMAINCTLSHMKEY
export RDBMBSWBOXNUM
export RDBFIXBUFSHMKEY
export RDBFIXBUFSHMADDR
export RDBFIXBUFSHMKEY2
export RDBFIXBUFSHMADDR2


timex rdbstart
```

******** 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 3
echo "   RDBII       "
#lcd 0 2 "RDB: closing now"
sleep 5
timex rdbstop
set rdbcheck = $status

#lcd 0 2 "(Ext.4197 Ozawa)"


if( $rdbcheck == 1 ) then
        echo "RDBSTOP:     : RDBII
ipcrm       "
        exit
endif

sleep 5

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
```

******** 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=/rdbptc/DireFile
RDBDIRSPACE2=/rdbptc/DireFile
RDBLOG=512, 512
#RDBLOG=256, 128
#RDBLOG=1024, 512
#RDBCORE=/rdbptc/RDBCORE
RDBCORE=/rdbptc/RDBCORE2
#RDBCNTNUM=256                  #fssqlenv ->
MAX_CONECT_SYS
RDBCNTNUM=350                  #fssqlenv ->
MAX_CONECT_SYS
RDBPRJCODE=0xdb

RDBSYSBUF=/rdbptc/tpcc80/SYS
RDBSQLENV=/rdbptc/tpcc80/SYS/fssqlenv
RDBLOGMANAGE=/rdbptc/tpcc80/SYS
RDBPOOLCFG=/rdbptc/tpcc80/SYS

#------------------------------------------------------------------
#RDBFIXBUFMEM=80            #LOAD
#RDBFIXBUFMEMADDR=0x36000000

###RDBFIXBUFMEM=2240          #98.06.01
###RDBFIXBUFMEMADDR=0x5f000000
#98.06.01
RDBFIXBUFMEM=2368          #98.06.14
RDBFIXBUFMEMADDR=0x57000000  #98.06.14


#-------------------------------------------------------
RDBEXTMEMADDR=0xed000000
RDBEXTMEM=4096

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

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

#####################################
#####################################nrk
##RDBVER:
UXP/DS_RDBII_V20L21_3/22_version
#####################################
#####################################
RDBSDPCPU = 1,2,3,4,5,6,7   ## SDP
CPU

RDBCCRDMCPU=0                       ##
daemon   CPU
RDBRECEPCPU=0
#---
RDBSORTCPU=0
RDBTCPIPCPU=0
RDBALFCPU=0
RDBDBSCPU=0          ##
RDBDIRCPU=0
#---
RDBIOCPU=0
RDBTLFCPU=0
#---
RDBWKSCPU=0


####I/O
RDBREADUNC = NO          # SAP
6/28
RDBDBSNUM+ = 190         #      /       '97/5/28
#RDBDBSNUM+ = 120         #LOAD
#RDBMAXLWP =  390       # DBSNUM
(  2)        LWPMAX=392
RDBMAXLWP =  350       #LOAD
RDBMAXDBIO = 20          # 20
I/O
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     #



******** 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=/rdbptc/DireFile
RDBDIRSPACE2=/rdbptc/DireFile
#RDBLOG=512, 512
RDBLOG=256, 128
#RDBLOG=1024, 512
RDBCORE=/rdbptc/RDBCORE
#RDBCNTNUM=256                  #fssqlenv ->
MAX_CONECT_SYS
RDBCNTNUM=350                  #fssqlenv ->
MAX_CONECT_SYS
RDBPRJCODE=0xdb

RDBSYSBUF=/rdbptc/tpcc80/SYS
RDBSQLENV=/rdbptc/tpcc80/SYS/fssqlenv
RDBLOGMANAGE=/rdbptc/tpcc80/SYS
RDBPOOLCFG=/rdbptc/tpcc80/SYS

#-------------------------------------------------------------------
RDBFIXBUFMEM=256             #LOAD
RDBFIXBUFMEMADDR=0x5f000000  #98.06.03
LOAD


#RDBFIXBUFMEM=2240
#RDBFIXBUFMEMADDR=0x61000000

#RDBFIXBUFMEM=1968        #MBS
#RDBFIXBUFMEM=1984        #MBS
##RDBFIXBUFMEM=2048        #MBS(+64)
##RDBFIXBUFMEMADDR=0x6b000000 #to
0xeb000000
#RDBFIXBUFMEM=2240          #98.06.01
#RDBFIXBUFMEMADDR=0x5f000000  #98.06.01


#-------------------------------------------------------------
#RDBEXTMEMADDR=0xed000000
RDBEXTMEMADDR=0x6f000000
RDBEXTMEM=4096

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

RDBKCHKSKIPCNT=200   #        IO
96.10.05
RDBKTAJUUDOSDP = 2016
RDBKLISTNUMSDP = 5
RDBKCATENUMSDP=252
#RDBKAIOREP = 6
RDBKAIOREP = 12  #980531
##RDBKAIOSAV = 30
RDBKAIOYLD = 0

RDBKAIOCNT    = yes
RDBKAIOD9F    = yes
RDBKAIOSELFWAIT    = yes
RDBKAIODSP    = yes

#####################################
#####################################nrk
##RDBVER:
UXP/DS_RDBII_V20L21_3/22_version
#####################################
#####################################
RDBSDPCPU = 0,1,2,3,4,5,6,7   ## SDP
CPU

RDBCCRDMCPU=0                       ##
daemon   CPU
RDBRECEPCPU=0
#---RDBSORTCPU=0
RDBTCPIPCPU=0
RDBALFCPU=0
RDBDBSCPU=0          ##
RDBDIRCPU=0
#---RDBIOCPU=0
RDBTLFCPU=0
#---RDBWKSCPU=0


####I/O
RDBREADUNC = NO          # SAP
6/28
RDBDBSNUM+ = 190         #      /       '97/5/28
#RDBDBSNUM+ = 120         #LOAD
###RDBMAXLWP =  390       # DBSNUM
(  2)        LWPMAX=392
RDBMAXLWP =  300       #350 LOAD
RDBMAXDBIO = 20          # 20
I/O
RDBMAXRCPIO = 20         # old=20 RCP
I/O
```

```
RDBNEWPAGE = 1          # LRU
#RDBNEWPAGE = 2          # LRU
RDBANTIQUEPAGE = 4 # LRU  4            ,
RDBIOUNITNUM = 32,1
RDBSORTUNITNUM = 32,1

##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     #



******** 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
******** sh.crbuf.mk.980714 ********
#!/bin/csh
rm sh.crbuf
rm mbs_tmp/*_*
echo "# 1200WH 2256MB##(Other
119MB)#############################"
>sh.crbuf

echo "# STOCK(781.2M->669MB)+924M----------
----------"  >>sh.crbuf

foreach P ( 1 2 3 4 5 6 7 8 9 10 11 12 13 14 )
 echo " rdbcrbf -A 10300 -S 10100 -x -I 1 -m 5
S_$P  4K 10700 ">>sh.crbuf
## echo " rdbcrbf -A 13500 -S 13300 -x -I 1 -m 5
S_$P  4K 13900 ">>sh.crbuf
end

#echo "0 21600">mbs_tmp/S_12

echo "0 49000">mbs_tmp/S_12
echo "0 49000">mbs_tmp/S_22
echo "1 49000">mbs_tmp/S_32
echo "1 49000">mbs_tmp/S_42
echo "2 49000">mbs_tmp/S_52
echo "2 49000">mbs_tmp/S_62
echo "3 49000">mbs_tmp/S_72
echo "3 49000">mbs_tmp/S_82
echo "4 49000">mbs_tmp/S_92
echo "4 49000">mbs_tmp/S_102
echo "5 49000">mbs_tmp/S_112
echo "5 49000">mbs_tmp/S_122
echo "6 49000">mbs_tmp/S_132
echo "6 49000">mbs_tmp/S_142

echo "# CUSTOMER(93.8M)----------
----------">>sh.crbuf
foreach P ( 1 2 3 4 5 6 7 )
# echo "rdbcrbf -A 700 -S 500 -x -I 1 -m 4 C_$P
8K 930">>sh.crbuf
echo "rdbcrbf -A 1660 -S 1460 -x -I 1 -m 4 C_$P
8K 1860">>sh.crbuf
end

echo "# CUSTOMER_IX(109.3M)----------
----------">>sh.crbuf
foreach P ( 1 2 3 4 5 6 7 )
 echo "rdbcrbf -A  1700  -S  1600 -x -I 1 -m 5
C_IX_$P  8K 1780">>sh.crbuf
end

echo "# ORDERS(78M)----------
------">>sh.crbuf
foreach P ( 1 2 3 4 5 6 7 )
 echo "rdbcrbf -A 1160  -S 1060 -x -I 1 -m 5 O_$P
8K 1200">>sh.crbuf
end

echo "# ORDER_IX(78M)+196M----------
----------">>sh.crbuf
foreach P ( 1 2 3 4 5 6 7 )
#echo "rdbcrbf -A  5880  -S  5700 -x -I 1 -m 5
O_IX_$P  4K 5980">>sh.crbuf
echo "rdbcrbf -A  7620  -S  7420 -x -I 1 -m 5
O_IX_$P  4K 7740">>sh.crbuf
end

echo "#ORDERLINE(468M)+672M----------
----------">>sh.crbuf
foreach P ( 1 2 3 4 5 6 7 8 9 10 11 12 13 14 )
# echo "rdbcrbf -A  3690  -S 3570 -x -I 1 -m 5
OL_$P  16K  3770">>sh.crbuf
  echo "rdbcrbf -A  1700  -S 1650 -x -I 1 -m 5
OL_$P  32K  1750">>sh.crbuf
end

echo "#ORDERLINE OVF----------
---">>sh.crbuf
foreach P ( 1 2 3 4 5 6 7 8 9 10 11 12 13 14 )
  echo "rdbcrbf -A  45  -S  35 -x -I 1 -m 5 OL_$P
8K  50">>sh.crbuf
end


#echo "#ORDERLINE_IX(198M)+280M----------
----------">>sh.crbuf
#foreach P ( 1 2 3 4 5 6 7 8 9 10 11 12 13 14 )
# echo "rdbcrbf -A   1850  -S  1810 -x -I 1 -m 5
OL_IX_$P  8K 1900">>sh.crbuf
#end

echo "#NEWORDER(128M)----------
----------">>sh.crbuf
foreach P ( 1 2 3 4 5 6 7 )
 echo "rdbcrbf -A 2260 -S  2210 -x -I 1 -m 5
NO_$P  8K 2310">>sh.crbuf
end
echo "#NEWORDER_IX(166MB)----------
----------">>sh.crbuf
foreach P ( 1 2 3 4 5 6 7 )
# echo "rdbcrbf -A   1875 -S   1850 -x -I 1 -m 5
NO_IX_$P  8K 1900">>sh.crbuf
  echo "rdbcrbf -A   5700 -S   5600 -x -I 1 -m 5
NO_IX_$P  8K 5800">>sh.crbuf
end
echo "#HISTORY(3M)----------
">>sh.crbuf
foreach P ( 1 2 3 4 5 6 7 )
 echo "rdbcrbf -A   130 -S   100 -x -I 1 -m 5 H_$P
4K 160">>sh.crbuf
end
```

```
******** 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
******** sh.rdbrcp.sub ********
#!/bin/sh
#
# rdbrcp       $1=No.
        date            >>res.doc/ckpt.$1
        echo 'RCP-START---'
>>res.doc/ckpt.$1
        date
        echo 'RCP-START----------------------
----------'
        timex rdbrcp
#       rdblog -V -a >> res.doc/ckpt.$1
#       rdblog -R -a
#       rdblog -R -a
#       rdblog -S -a
#       rdblog -R -a  #8H RUN
        echo 'RCP-END------------------------
----------'
        date
        date            >>res.doc/ckpt.$1
        echo 'RCP-END---'
>>res.doc/ckpt.$1
        echo ' '        >>res.doc/ckpt.$1

******** sql.env ********
SERVER_SPEC = ( RDB2_TCP, SV1 ,TPCC ,
tpc225 , 2001 )
DEFAULT_CONNECTION = (
TPCC.TPCC,tpcc,tpcc )
TRAN_SPEC = ( TRANSACTION_ROLLBACK )
DESCRIPTOR_SPEC = ( 30,1 )
;BUFFER_SIZE = ( 16 )
WAIT_TIME = ( 0 )
NCHAR_CODE = ( EUC )
OPL_BUFFER_SIZE = ( 280 )
;;;CHARACTER_TRANSLATE = CLIENT
CHAR_CODE = EUC
RESULT_BUFFER = ( 4,1 )
;SQL_SNAP =
(ON,/risu02/tpcc.tcp.snap.970206,2 )
;MSG_PRINT = ( ON )
DSO_LOCK =
(TPCC.ORDERLINE_DSO/EX,TPCC.HISTORY_
DSO/EX,TPCC.CUSTOMER_IX_DSO/SH,

TPCC.ITEM_DSO/SH,TPCC.NEWORDER_DSO/
EX,TPCC.ORDERS_DSO/EX,

TPCC.NEWORDER_IX_DSO/EX,TPCC.ORDER
S_IX_DSO/EX)
;DSO_LOCK = (
TPCC.CUSTOMER_IX_DSO/SH,TPCC.ITEM_DS
O/SH )
SIGNAL_INF = NO
SORT_MEM_SIZE = 128
WORK_MEM_SIZE = 64
******** system ********
*ident          "@(#)system             1.15
        92/11/14 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@0,8
00000/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 = 12288
set semsys:seminfo_semmns = 9216
set semsys:seminfo_semmnu = 3072

*set shmsys:shminfo_shmmax = 104857600
set shmsys:shminfo_shmmax = 2097152000
set shmsys:shminfo_shmmni = 1024
set shmsys:shminfo_shmseg = 512

* TCP-IP Max User
set pt_cnt = 200
* vxvm_START (do not remove)
forceload: drv/vxdmp
forceload: drv/vxio
forceload: drv/vxspec
* vxvm_END (do not remove)

set  enable_grp_ism = 1
******** system.client ********
*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@0,8
00000/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 pt_cnt = 3500

set shmsys:shminfo_shmmax=268435456
set semsys:seminfo_semume=4000
set semsys:seminfo_semmnu=4000
set semsys:seminfo_semmns=4000
set semsys:seminfo_semmsl=500
set semsys:seminfo_semmni=100
set semsys:seminfo_semmap=100
set msgsys:msginfo_msgmni=4000
set msgsys:msginfo_msgmap=200
set msgsys:msginfo_msgmax=16384
set msgsys:msginfo_msgmnb=16384
set msgsys:msginfo_msgtql=4000
set msgsys:msginfo_msgseg=32767
set msgsys:msginfo_msgssz=128

* set tune_t_fsflushr=20
* set autoup=600
******** ubbconfig ********
#
#       ubbconfig : TUXEDO configuration file
#

*RESOURCES
IPCKEY          90952
MASTER          SITE1
UID             100
GID             100
PERM            0660
MAXACCESSERS    2250
MAXSERVERS      50
MAXSERVICES     2200
MODEL           SHM
LDBAL           Y
BLOCKTIME               30

*MACHINES
cl04                    LMID=SITE1
        TUXCONFIG="/home/tpcc/tuxconfig"
        ROOTDIR="/opt/FSUNtpbs"
        APPDIR="/home/tpcc/APL.fml"
        ULOGPFX="/home/tpcc/log/ULOG"

ENVFILE="/home/tpcc/APL.fml/txenv.tcp"

*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


*SERVERS
DEFAULT:        RESTART=Y MAXGEN=5
REPLYQ=N RQPERM=0660
TPCC    SRVGRP=group1 RQADDR=TPCCq1
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group2 RQADDR=TPCCq2
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group3 RQADDR=TPCCq3
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group4 RQADDR=TPCCq4
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group5 RQADDR=TPCCq5
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group6 RQADDR=TPCCq6
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group7 RQADDR=TPCCq7
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group8 RQADDR=TPCCq8
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group9 RQADDR=TPCCq9
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group10 RQADDR=TPCCq10
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group11 RQADDR=TPCCq11
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group12 RQADDR=TPCCq12
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group13 RQADDR=TPCCq13
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group14 RQADDR=TPCCq14
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group15 RQADDR=TPCCq15
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group16 RQADDR=TPCCq16
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group17 RQADDR=TPCCq17
SRVID=1 CLOPT="-s TPCC:TPCC"
```

```
TPCC    SRVGRP=group18 RQADDR=TPCCq18
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group19 RQADDR=TPCCq19
SRVID=1 CLOPT="-s TPCC:TPCC"
TPCC    SRVGRP=group20 RQADDR=TPCCq20
SRVID=1 CLOPT="-s TPCC:TPCC"


# *NETWORK

*SERVICES
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group1
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group2
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group3
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group4
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group5
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group6
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group7
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group8
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group9
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group10
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group11
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group12
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group13
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group14
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group15
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group16
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group17
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group18
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group19
"TPCC"  TRANTIME=0     ROUTING="route1"
SRVGRP=group20


*ROUTING
"route1"       FIELD=FML_TERM
            BUFTYPE="FML"
                           RANGES="1-
10:group1,11-20:group2,21-30:group3,31-
40:group4,41-50:group5,51-60:group6,61-
70:group7,71-80:group8,81-90:group9,91-
100:group10,101-110:group11,111-
120:group12,121-130:group13,131-
140:group14,141-150:group15,151-
160:group16,161-170:group17,171-
180:group18,181-190:group19,191-
200:group20,*:*"

********  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

```
******** ALL.mk ********
:
set -x

RDB_FAST_OPEN=1
export RDB_FAST_OPEN

CRDIC

date
RDBSTART.ddl
rdbddlex ddl_db.mak
rdbddlex -d TPCC crta.def.dec_to_int

rdbddlex -d TPCC ddl.dbsp.dat.p1

sleep 5
RDBSTOP
exit

RDBSTART.ddl

rdbddlex -d TPCC ddl.dbsp.dat.p2

sleep 5
RDBSTOP
RDBSTART.ddl

rdbddlex -d TPCC ddl.dat.WH
rdbddlex -d TPCC ddl.dat.DI
rdbddlex -d TPCC ddl.dat.ST
rdbddlex -d TPCC ddl.dat.NO
rdbddlex -d TPCC ddl.dat.NI
rdbddlex -d TPCC ddl.dat.OS
rdbddlex -d TPCC ddl.dat.OI
rdbddlex -d TPCC ddl.dat.HI
rdbddlex -d TPCC ddl.dat.CU
rdbddlex -d TPCC ddl.dat.CI
rdbddlex -d TPCC ddl.dat.OL
rdbddlex -d TPCC ddl.dat.IT
sleep 5
RDBSTOP
date

RDBSTART.ddl
stored.sh
sleep 5
RDBSTOP


RDBSTART.ddl
timex csh -x LOAD.sh
RDBSTOP

sleep 5

sh mktmplog.sh
sh mkarc.sh
```

```
******** CRDIC ********
:
set -x
SYS=/rdbptc/tpcc80/SYS
LOG_IX=/dev/vx/rdsk/vola03_02
LOG_BI=/dev/vx/rdsk/vola03_01
LOG_AI=/dev/vx/rdsk/vola01_01
DIC_PL=/dev/rdsk/c2t64d0s4
BISZ=64M
AISZ=64M
TRN=100
RCV=32M

date
rm -f /rdb/loaddata/32/RDBLOG/LOG_AI
rm -f $SYS/rdblogmanage

rdblog -l
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
******** crta.def.dec_to_int ********
--
#####################################
#####################################
-- #          table definition of TPC-C model
-- #
-- #          1995.5.15 Arakawa
-- #          1996.4.18 Muramatsu
           DECIMAL -> CHAR or SMALLINT or
INTEGER
-- #          1996.10.18 Hara
           C_ID,H_C_ID,O_C_ID SMALLINT ->
INTEGER
-- #
                          I_IM_ID
-- #
--
#####################################
#####################################
CREATE SCHEMA TPCC_SCHEMA
--
#####################################
#####################################
-- # TABLE definition
--
#####################################
#####################################
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             SMALLINT
          NOT NULL,
          W_YTD
          DECIMAL(12,2) 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             SMALLINT
          NOT NULL,
          D_YTD
          DECIMAL(12,2)     NOT NULL,
          D_NEXT_O_ID       INTEGER
          NOT NULL,
          PRIMARY KEY(D_W_ID,D_ID)
)

CREATE TABLE TPCC_SCHEMA.CUSTOMER(
          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           CHAR(14)
          NOT NULL,
          C_CREDIT CHAR(2)
          NOT NULL,
          C_CREDIT_LIM
          DECIMAL(12,2)     NOT NULL,
          C_DISCOUNT        SMALLINT
          NOT NULL,
          C_BALANCE
          DECIMAL(12,2)     NOT NULL,
          C_YTD_PAYMENT
          DECIMAL(12,2)     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             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              INTEGER
        NOT NULL,
        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       CHAR(14),
        OL_QUANTITY         SMALLINT
        NOT NULL,
        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)
)

CREATE TABLE TPCC_SCHEMA.HISTORY(
        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              CHAR(14)
        NOT NULL,
        H_AMOUNT            INTEGER
        NOT NULL,
        H_DATA              CHAR(24)
        NOT NULL
)
******** ddl.dat.CI ********
-------------------------------------------------------------------
---------
-- * 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 7648K;

        CREATE DSI CUSTOMER_X_2_DSI
                INDEX
                DSO
CUSTOMER_IX_DSO
                BASE
CUSTOMER_2_DSI
                ALLOCATE INDEX   ON
SP2 SIZE 64K,
                    BASE    ON SP2
SIZE 7648K;




        CREATE DSI
CUSTOMER_X_167_DSI
                INDEX
                DSO
CUSTOMER_IX_DSO
                BASE
CUSTOMER_167_DSI
                ALLOCATE INDEX   ON
SP167 SIZE 64K,
                    BASE    ON SP167
SIZE 7648K;

        CREATE DSI
CUSTOMER_X_168_DSI
                INDEX
                DSO
CUSTOMER_IX_DSO
                BASE
CUSTOMER_168_DSI
                ALLOCATE INDEX   ON
SP168 SIZE 64K,
                    BASE    ON SP168
SIZE 7648K;

******** ddl.dat.CU ********
-------------------------------------------------------------------
---------
-- * Phase.2-3a: Customer
-------------------------------------------------------------------
---------
        CREATE DSO CUSTOMER_DSO
                FROM
TPCC_SCHEMA.CUSTOMER
                    TYPE
RANDOM(PAGESIZE1(8),PAGESIZE2(1),

        RULE((C_ID*10+C_D_ID+(C_W_ID-
(C_W_ID/10*10))*30000)))
                WHERE
(C_W_ID) BETWEEN (?) AND (?);

        CREATE DSI CUSTOMER_1_DSI
            DSO CUSTOMER_DSO
                USING(1,10)
                ALLOCATE PRIME   ON
SP1 SIZE 240008K,
                    OVERFLOW ON
SP1 SIZE  12002K;
```

```
            CREATE DSI CUSTOMER_2_DSI
                DSO CUSTOMER_DSO
                        USING(11,20)
                        ALLOCATE PRIME    ON
SP2 SIZE 240008K,
                                OVERFLOW ON
SP2 SIZE  12002K;




            CREATE DSI CUSTOMER_167_DSI
                DSO CUSTOMER_DSO
                        USING(1661,1670)
                        ALLOCATE PRIME    ON
SP167 SIZE 240008K,
                                OVERFLOW ON
SP167 SIZE  12002K;

            CREATE DSI CUSTOMER_168_DSI
                DSO CUSTOMER_DSO
                        USING(1671,3360)
                        ALLOCATE PRIME    ON
SP168 SIZE 240008K,
                                OVERFLOW ON
SP168 SIZE  12002K;

******** ddl.dat.DI ********
-----------------------------------------------------------------------
---------
-- * 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(1,60)
                        ALLOCATE PRIME    ON
SP29 SIZE 1201K,
                                OVERFLOW ON
SP29 SIZE 30K;

            CREATE DSI DISTRICT_2_DSI
                DSO DISTRICT_DSO
                        USING(61,120)
                        ALLOCATE PRIME    ON
SP30 SIZE 1201K,
                                OVERFLOW ON
SP30 SIZE 30K;




            CREATE DSI DISTRICT_27_DSI
                DSO DISTRICT_DSO
                        USING(1561,1620)
                        ALLOCATE PRIME    ON
SP55 SIZE 1201K,
                                OVERFLOW ON
SP55 SIZE 30K;

            CREATE DSI DISTRICT_28_DSI
                DSO DISTRICT_DSO
                        USING(1621,3360)
                        ALLOCATE PRIME    ON
SP56 SIZE 1201K,
                                OVERFLOW ON
SP56 SIZE 30K;

******** ddl.dat.HI ********
-----------------------------------------------------------------------
---------
-- * 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(1,10)
                        ALLOCATE DATA     ON
SP1 SIZE 29492K;

            CREATE DSI HISTORY_2_DSI
                DSO HISTORY_DSO
                        USING(11,20)
                        ALLOCATE DATA     ON
SP2 SIZE 29492K;




            CREATE DSI HISTORY_167_DSI
                DSO HISTORY_DSO
                        USING(1661,1670)
                        ALLOCATE DATA     ON
SP167 SIZE 29492K;

            CREATE DSI HISTORY_168_DSI
                DSO HISTORY_DSO
                        USING(1671,3360)
                        ALLOCATE DATA     ON
SP168 SIZE 29492K;

******** ddl.dat.IT ********
-----------------------------------------------------------------------
---------
-- * 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 1435K
                SP2 SIZE 1428K

                SP3 SIZE 1428K

                SP4 SIZE 1428K

                SP5 SIZE 1428K

                SP6 SIZE 1428K

                SP7 SIZE 1428K

                SP8 SIZE 1428K

                SP9 SIZE 1428K

                SP10 SIZE 1428K,
                        OVERFLOW ON
                SP11 SIZE 716K;

******** ddl.dat.NI ********
-----------------------------------------------------------------------
---------
-- * 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),DEGEN
ERATE);

            CREATE DSI NEWORDER_X_1_DSI
                        INDEX
                        DSO
NEWORDER_IX_DSO
                        BASE
NEWORDER_1_DSI
                        ALLOCATE INDEX    ON
SP1 SIZE 128K,
                                BASE    ON SP1
SIZE 6016K;

            CREATE DSI NEWORDER_X_2_DSI
                        INDEX
                        DSO
NEWORDER_IX_DSO
                        BASE
NEWORDER_2_DSI
                        ALLOCATE INDEX    ON
SP2 SIZE 128K,
                                BASE    ON SP2
SIZE 6016K;
```

```
            CREATE DSI
NEWORDER_X_167_DSI
                        INDEX
                        DSO
NEWORDER_IX_DSO
                        BASE
NEWORDER_167_DSI
                        ALLOCATE INDEX    ON
SP167 SIZE 128K,
                                BASE    ON SP167
SIZE 6016K;

            CREATE DSI
NEWORDER_X_168_DSI
                        INDEX
                        DSO
NEWORDER_IX_DSO
                        BASE
NEWORDER_168_DSI
                        ALLOCATE INDEX    ON
SP168 SIZE 128K,
                                BASE    ON SP168
SIZE 6016K;

******** ddl.dat.NO ********
-------------------------------------------------------------------------
---------
-- *Phase.2-6a: NewOrder
-------------------------------------------------------------------------
---------
            CREATE DSO NEWORDER_DSO
                        FROM
TPCC_SCHEMA.NEWORDER
        TYPE
RANDOM(PAGESIZE1(8),PAGESIZE2(1),RULE((
NO_O_ID/8)*10+NO_W_ID+((NO_D_ID-
1)*10+(NO_O_ID-((NO_O_ID/8)*8)))*1630))
                                    WHERE
(NO_W_ID) BETWEEN (?) AND (?);

            CREATE DSI NEWORDER_1_DSI
                DSO NEWORDER_DSO
                        USING(1,10)
                        ALLOCATE PRIME    ON
SP1 SIZE 13048K,
                                OVERFLOW ON
SP1 SIZE 456K;

            CREATE DSI NEWORDER_2_DSI
                DSO NEWORDER_DSO
                        USING(11,20)
                        ALLOCATE PRIME    ON
SP2 SIZE 13048K,
                                OVERFLOW ON
SP2 SIZE 456K;




            CREATE DSI NEWORDER_167_DSI
                DSO NEWORDER_DSO
                        USING(1661,1670)
                        ALLOCATE PRIME    ON
SP167 SIZE 13048K,
                                OVERFLOW ON
SP167 SIZE 456K;

            CREATE DSI NEWORDER_168_DSI
                        DSO NEWORDER_DSO
                                USING(1671,3360)
                                ALLOCATE PRIME    ON
SP168 SIZE 13048K,
                                        OVERFLOW ON
SP168 SIZE 456K;

******** ddl.dat.OI ********
-------------------------------------------------------------------------
---------
-- *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 14912K;

            CREATE DSI ORDERS_IX_2_DSI
                        INDEX
                        DSO ORDERS_IX_DSO
                        BASE ORDERS_2_DSI
                        ALLOCATE INDEX    ON
SP2 SIZE 512K,
                                BASE    ON SP2
SIZE 14912K;




            CREATE DSI ORDERS_IX_167_DSI
                        INDEX
                        DSO ORDERS_IX_DSO
                        BASE
ORDERS_167_DSI
                        ALLOCATE INDEX    ON
SP167 SIZE 512K,
                                BASE    ON SP167
SIZE 14912K;

            CREATE DSI ORDERS_IX_168_DSI
                        INDEX
                        DSO ORDERS_IX_DSO
                        BASE
ORDERS_168_DSI
                        ALLOCATE INDEX    ON
SP168 SIZE 512K,
                                BASE    ON SP168
SIZE 14912K;

******** ddl.dat.OL ********
-------------------------------------------------------------------------
---------
-- *Phase.2-5a: OrderLine
-------------------------------------------------------------------------
---------
            CREATE DSO ORDERLINE_DSO
                        FROM
TPCC_SCHEMA.ORDERLINE
                        TYPE
RANDOM(PAGESIZE1(32),PAGESIZE2(8),RULE
((OL_O_ID/30)*100+OL_W_ID*10+OL_D_ID+(OL
_NUMBER+(OL_O_ID-
((OL_O_ID/30)*30))*15)*12505))
                        WHERE
(OL_W_ID) BETWEEN (?) AND (?);

            CREATE DSI ORDERLIN_1_DSI
                DSO ORDERLINE_DSO
                        USING(1,10)
                        ALLOCATE PRIME    ON
SP1 SIZE 400192K,
                                OVERFLOW ON
SP1 SIZE  1600K;

            CREATE DSI ORDERLIN_2_DSI
                DSO ORDERLINE_DSO
                        USING(11,20)
                        ALLOCATE PRIME    ON
SP2 SIZE 400192K,
                                OVERFLOW ON
SP2 SIZE  1600K;




            CREATE DSI ORDERLIN_167_DSI
                DSO ORDERLINE_DSO
                        USING(1661,1670)
                        ALLOCATE PRIME    ON
SP167 SIZE 400192K,
                                OVERFLOW ON
SP167 SIZE  1600K;

            CREATE DSI ORDERLIN_168_DSI
                DSO ORDERLINE_DSO
                        USING(1671,3360)
                        ALLOCATE PRIME    ON
SP168 SIZE 400192K,
                                OVERFLOW ON
SP168 SIZE  1600K;

******** ddl.dat.OS ********
-------------------------------------------------------------------------
---------
-- *Phase.2-4a: Orders
-------------------------------------------------------------------------
---------
            CREATE DSO ORDERS_DSO
                        FROM
TPCC_SCHEMA.ORDERS
                        TYPE
RANDOM(PAGESIZE1(8),PAGESIZE2(1),RULE((
O_ID/8)*10+O_W_ID+((O_D_ID-1)*10+(O_ID-
((O_ID/8)*8)))*4769))
                        WHERE
(O_W_ID) BETWEEN (?) AND (?);

            CREATE DSI ORDERS_1_DSI
                DSO ORDERS_DSO
                        USING(1,10)
                        ALLOCATE PRIME    ON
SP1 SIZE 38160k,
                                OVERFLOW ON
SP1 SIZE   301K;
```

```
        CREATE DSI ORDERS_2_DSI
            DSO ORDERS_DSO
                USING(11,20)
                ALLOCATE PRIME    ON
SP2 SIZE 38160k,
                    OVERFLOW ON
SP2 SIZE   301K;



        CREATE DSI ORDERS_167_DSI
            DSO ORDERS_DSO
                USING(1661,1670)
                ALLOCATE PRIME    ON
SP167 SIZE 38160k,
                    OVERFLOW ON
SP167 SIZE   301K;

        CREATE DSI ORDERS_168_DSI
            DSO ORDERS_DSO
                USING(1671,3360)
                ALLOCATE PRIME    ON
SP168 SIZE 38160k,
                    OVERFLOW ON
SP168 SIZE   301K;

******** ddl.dat.ST ********
-------------------------------------------------------------------
---------
-- *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)/10+(S_W_ID-S_W_ID/10*10)*300000))
                    WHERE
(S_W_ID) BETWEEN (?) AND (?);

        CREATE DSI STOCK_1_DSI
            DSO STOCK_DSO
                USING(1,30)
                ALLOCATE PRIME    ON
SP1 SIZE 200004K

                                  SP2 SIZE
200000K

                                  SP3 SIZE
200000K

                                  SP4 SIZE
200000K

                                  SP5 SIZE
200000K

                                  SP6 SIZE
200000K,
                    OVERFLOW ON
SP1 SIZE 60002K;

        CREATE DSI STOCK_2_DSI
            DSO STOCK_DSO
                USING(31,60)
                ALLOCATE PRIME    ON
SP1 SIZE 200004K

                                  SP2 SIZE
200000K

                                  SP3 SIZE
200000K

                                  SP4 SIZE
200000K

                                  SP5 SIZE
200000K

                                  SP6 SIZE
200000K,
                    OVERFLOW ON
SP2 SIZE 60002K;

        CREATE DSI STOCK_3_DSI
            DSO STOCK_DSO
                USING(61,90)
                ALLOCATE PRIME    ON
SP7 SIZE 200004K

                                  SP8 SIZE
200000K

                                  SP9 SIZE
200000K

                                  SP10 SIZE
200000K

                                  SP11 SIZE
200000K

                                  SP12 SIZE
200000K,
                    OVERFLOW ON
SP7 SIZE 60002K;

        CREATE DSI STOCK_4_DSI
            DSO STOCK_DSO
                USING(91,120)
                ALLOCATE PRIME    ON
SP7 SIZE 200004K

                                  SP8 SIZE
200000K

                                  SP9 SIZE
200000K

                                  SP10 SIZE
200000K

                                  SP11 SIZE
200000K

                                  SP12 SIZE
200000K,
                    OVERFLOW ON
SP8 SIZE 60002K;

        CREATE DSI STOCK_55_DSI
            DSO STOCK_DSO
                USING(1621,1650)
                ALLOCATE PRIME    ON
SP163 SIZE 200004K

                                  SP164 SIZE
200000K

                                  SP165 SIZE
200000K

                                  SP166 SIZE
200000K

                                  SP167 SIZE
200000K

                                  SP168 SIZE
200000K,
                    OVERFLOW ON
SP163 SIZE 60002K;

        CREATE DSI STOCK_56_DSI
            DSO STOCK_DSO
                USING(1651,3360)
                ALLOCATE PRIME    ON
SP163 SIZE 200004K

                                  SP164 SIZE
200000K

                                  SP165 SIZE
200000K

                                  SP166 SIZE
200000K

                                  SP167 SIZE
200000K

                                  SP168 SIZE
200000K,
                    OVERFLOW ON
SP164 SIZE 60002K;

******** ddl.dat.WH ********
-------------------------------------------------------------------
---------
-- *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(1,60)
                ALLOCATE PRIME    ON
SP1 SIZE 3096K,
                    OVERFLOW ON
SP1 SIZE 12K;
```

```
        CREATE DSI WAREHOUSE_2_DSI
            DSO WAREHOUSE_DSO
                USING(61,120)
                ALLOCATE PRIME   ON
SP2 SIZE 3096K,
                        OVERFLOW ON
SP2 SIZE 12K;




        CREATE DSI WAREHOUSE_27_DSI
            DSO WAREHOUSE_DSO
                USING(1561,1620)
                ALLOCATE PRIME   ON
SP27 SIZE 3096K,
                        OVERFLOW ON
SP27 SIZE 12K;

        CREATE DSI WAREHOUSE_28_DSI
            DSO WAREHOUSE_DSO
                USING(1621,3360)
                ALLOCATE PRIME   ON
SP28 SIZE 3096K,
                        OVERFLOW ON
SP28 SIZE 12K;

******** ddl.dbsp.dat.p1 ********
CREATE DBSPACE SP1   ALLOCATE
RAWDEVICE /dev/rdsk/c2t64d0s1;
CREATE DBSPACE SP2   ALLOCATE
RAWDEVICE /dev/rdsk/c6t65d0s1;
CREATE DBSPACE SP3   ALLOCATE
RAWDEVICE /dev/rdsk/c2t66d0s1;
CREATE DBSPACE SP4   ALLOCATE
RAWDEVICE /dev/rdsk/c21t48d0s1;
CREATE DBSPACE SP5   ALLOCATE
RAWDEVICE /dev/rdsk/c2t68d0s1;
CREATE DBSPACE SP6   ALLOCATE
RAWDEVICE /dev/rdsk/c6t69d0s1;
CREATE DBSPACE SP7   ALLOCATE
RAWDEVICE /dev/rdsk/c2t70d0s1;
CREATE DBSPACE SP8   ALLOCATE
RAWDEVICE /dev/rdsk/c6t80d0s1;
CREATE DBSPACE SP9   ALLOCATE
RAWDEVICE /dev/rdsk/c2t81d0s1;
CREATE DBSPACE SP10  ALLOCATE
RAWDEVICE /dev/rdsk/c6t82d0s1;
CREATE DBSPACE SP11  ALLOCATE
RAWDEVICE /dev/rdsk/c2t83d0s1;
CREATE DBSPACE SP12  ALLOCATE
RAWDEVICE /dev/rdsk/c6t84d0s1;
CREATE DBSPACE SP13  ALLOCATE
RAWDEVICE /dev/rdsk/c2t85d0s1;
CREATE DBSPACE SP14  ALLOCATE
RAWDEVICE /dev/rdsk/c6t86d0s1;
CREATE DBSPACE SP15  ALLOCATE
RAWDEVICE /dev/rdsk/c2t96d0s1;
CREATE DBSPACE SP16  ALLOCATE
RAWDEVICE /dev/rdsk/c6t97d0s1;
CREATE DBSPACE SP17  ALLOCATE
RAWDEVICE /dev/rdsk/c2t98d0s1;
CREATE DBSPACE SP18  ALLOCATE
RAWDEVICE /dev/rdsk/c6t99d0s1;
CREATE DBSPACE SP19  ALLOCATE
RAWDEVICE /dev/rdsk/c2t100d0s1;
CREATE DBSPACE SP20  ALLOCATE
RAWDEVICE /dev/rdsk/c6t101d0s1;
CREATE DBSPACE SP21  ALLOCATE
RAWDEVICE /dev/rdsk/c2t102d0s1;
CREATE DBSPACE SP22  ALLOCATE
RAWDEVICE /dev/rdsk/c6t112d0s1;
CREATE DBSPACE SP23  ALLOCATE
RAWDEVICE /dev/rdsk/c2t113d0s1;
CREATE DBSPACE SP24  ALLOCATE
RAWDEVICE /dev/rdsk/c6t114d0s1;
CREATE DBSPACE SP25  ALLOCATE
RAWDEVICE /dev/rdsk/c2t115d0s1;
CREATE DBSPACE SP26  ALLOCATE
RAWDEVICE /dev/rdsk/c6t116d0s1;
CREATE DBSPACE SP27  ALLOCATE
RAWDEVICE /dev/rdsk/c2t117d0s1;
CREATE DBSPACE SP28  ALLOCATE
RAWDEVICE /dev/rdsk/c6t118d0s1;
CREATE DBSPACE SP29  ALLOCATE
RAWDEVICE /dev/rdsk/c10t0d0s1;
CREATE DBSPACE SP30  ALLOCATE
RAWDEVICE /dev/rdsk/c11t1d0s1;
CREATE DBSPACE SP31  ALLOCATE
RAWDEVICE /dev/rdsk/c10t2d0s1;
CREATE DBSPACE SP32  ALLOCATE
RAWDEVICE /dev/rdsk/c11t3d0s1;
CREATE DBSPACE SP33  ALLOCATE
RAWDEVICE /dev/rdsk/c10t4d0s1;
CREATE DBSPACE SP34  ALLOCATE
RAWDEVICE /dev/rdsk/c11t5d0s1;
CREATE DBSPACE SP35  ALLOCATE
RAWDEVICE /dev/rdsk/c10t6d0s1;
CREATE DBSPACE SP36  ALLOCATE
RAWDEVICE /dev/rdsk/c11t16d0s1;
CREATE DBSPACE SP37  ALLOCATE
RAWDEVICE /dev/rdsk/c10t17d0s1;
CREATE DBSPACE SP38  ALLOCATE
RAWDEVICE /dev/rdsk/c11t18d0s1;
CREATE DBSPACE SP39  ALLOCATE
RAWDEVICE /dev/rdsk/c10t19d0s1;
CREATE DBSPACE SP40  ALLOCATE
RAWDEVICE /dev/rdsk/c11t20d0s1;
CREATE DBSPACE SP41  ALLOCATE
RAWDEVICE /dev/rdsk/c10t21d0s1;
CREATE DBSPACE SP42  ALLOCATE
RAWDEVICE /dev/rdsk/c11t22d0s1;
CREATE DBSPACE SP43  ALLOCATE
RAWDEVICE /dev/rdsk/c10t32d0s1;
CREATE DBSPACE SP44  ALLOCATE
RAWDEVICE /dev/rdsk/c11t33d0s1;
CREATE DBSPACE SP45  ALLOCATE
RAWDEVICE /dev/rdsk/c10t34d0s1;
CREATE DBSPACE SP46  ALLOCATE
RAWDEVICE /dev/rdsk/c11t35d0s1;
CREATE DBSPACE SP47  ALLOCATE
RAWDEVICE /dev/rdsk/c10t36d0s1;
CREATE DBSPACE SP48  ALLOCATE
RAWDEVICE /dev/rdsk/c11t37d0s1;
CREATE DBSPACE SP49  ALLOCATE
RAWDEVICE /dev/rdsk/c10t38d0s1;
CREATE DBSPACE SP50  ALLOCATE
RAWDEVICE /dev/rdsk/c11t48d0s1;
CREATE DBSPACE SP51  ALLOCATE
RAWDEVICE /dev/rdsk/c10t49d0s1;
CREATE DBSPACE SP52  ALLOCATE
RAWDEVICE /dev/rdsk/c11t50d0s1;
CREATE DBSPACE SP53  ALLOCATE
RAWDEVICE /dev/rdsk/c10t51d0s1;
CREATE DBSPACE SP54  ALLOCATE
RAWDEVICE /dev/rdsk/c11t52d0s1;
CREATE DBSPACE SP55  ALLOCATE
RAWDEVICE /dev/rdsk/c10t53d0s1;
CREATE DBSPACE SP56  ALLOCATE
RAWDEVICE /dev/rdsk/c11t54d0s1;
CREATE DBSPACE SP57  ALLOCATE
RAWDEVICE /dev/rdsk/c16t64d0s1;
CREATE DBSPACE SP58  ALLOCATE
RAWDEVICE /dev/rdsk/c17t65d0s1;
CREATE DBSPACE SP59  ALLOCATE
RAWDEVICE /dev/rdsk/c16t66d0s1;
CREATE DBSPACE SP60  ALLOCATE
RAWDEVICE /dev/rdsk/c17t67d0s1;
CREATE DBSPACE SP61  ALLOCATE
RAWDEVICE /dev/rdsk/c16t68d0s1;
CREATE DBSPACE SP62  ALLOCATE
RAWDEVICE /dev/rdsk/c17t69d0s1;
CREATE DBSPACE SP63  ALLOCATE
RAWDEVICE /dev/rdsk/c16t70d0s1;
CREATE DBSPACE SP64  ALLOCATE
RAWDEVICE /dev/rdsk/c17t80d0s1;
CREATE DBSPACE SP65  ALLOCATE
RAWDEVICE /dev/rdsk/c16t81d0s1;
CREATE DBSPACE SP66  ALLOCATE
RAWDEVICE /dev/rdsk/c17t82d0s1;
CREATE DBSPACE SP67  ALLOCATE
RAWDEVICE /dev/rdsk/c16t83d0s1;
CREATE DBSPACE SP68  ALLOCATE
RAWDEVICE /dev/rdsk/c17t84d0s1;
CREATE DBSPACE SP69  ALLOCATE
RAWDEVICE /dev/rdsk/c16t85d0s1;
CREATE DBSPACE SP70  ALLOCATE
RAWDEVICE /dev/rdsk/c17t86d0s1;
CREATE DBSPACE SP71  ALLOCATE
RAWDEVICE /dev/rdsk/c16t96d0s1;
CREATE DBSPACE SP72  ALLOCATE
RAWDEVICE /dev/rdsk/c17t97d0s1;
CREATE DBSPACE SP73  ALLOCATE
RAWDEVICE /dev/rdsk/c16t98d0s1;
CREATE DBSPACE SP74  ALLOCATE
RAWDEVICE /dev/rdsk/c17t99d0s1;
CREATE DBSPACE SP75  ALLOCATE
RAWDEVICE /dev/rdsk/c16t100d0s1;
CREATE DBSPACE SP76  ALLOCATE
RAWDEVICE /dev/rdsk/c17t101d0s1;
******** ddl.dbsp.dat.p2 ********
CREATE DBSPACE SP77  ALLOCATE
RAWDEVICE /dev/rdsk/c16t102d0s1;
CREATE DBSPACE SP78  ALLOCATE
RAWDEVICE /dev/rdsk/c17t112d0s1;
CREATE DBSPACE SP79  ALLOCATE
RAWDEVICE /dev/rdsk/c16t113d0s1;
CREATE DBSPACE SP80  ALLOCATE
RAWDEVICE /dev/rdsk/c17t114d0s1;
CREATE DBSPACE SP81  ALLOCATE
RAWDEVICE /dev/rdsk/c16t115d0s1;
CREATE DBSPACE SP82  ALLOCATE
RAWDEVICE /dev/rdsk/c17t116d0s1;
CREATE DBSPACE SP83  ALLOCATE
RAWDEVICE /dev/rdsk/c16t117d0s1;
CREATE DBSPACE SP84  ALLOCATE
RAWDEVICE /dev/rdsk/c17t118d0s1;
CREATE DBSPACE SP85  ALLOCATE
RAWDEVICE /dev/rdsk/c13t0d0s1;
CREATE DBSPACE SP86  ALLOCATE
RAWDEVICE /dev/rdsk/c13t1d0s1;
CREATE DBSPACE SP87  ALLOCATE
RAWDEVICE /dev/rdsk/c13t2d0s1;
CREATE DBSPACE SP88  ALLOCATE
RAWDEVICE /dev/rdsk/c13t3d0s1;
```

```
CREATE DBSPACE SP89  ALLOCATE
RAWDEVICE /dev/rdsk/c13t4d0s1;
CREATE DBSPACE SP90  ALLOCATE
RAWDEVICE /dev/rdsk/c13t5d0s1;
CREATE DBSPACE SP91  ALLOCATE
RAWDEVICE /dev/rdsk/c13t6d0s1;
CREATE DBSPACE SP92  ALLOCATE
RAWDEVICE /dev/rdsk/c13t16d0s1;
CREATE DBSPACE SP93  ALLOCATE
RAWDEVICE /dev/rdsk/c13t17d0s1;
CREATE DBSPACE SP94  ALLOCATE
RAWDEVICE /dev/rdsk/c13t18d0s1;
CREATE DBSPACE SP95  ALLOCATE
RAWDEVICE /dev/rdsk/c13t19d0s1;
CREATE DBSPACE SP96  ALLOCATE
RAWDEVICE /dev/rdsk/c13t20d0s1;
CREATE DBSPACE SP97  ALLOCATE
RAWDEVICE /dev/rdsk/c13t21d0s1;
CREATE DBSPACE SP98  ALLOCATE
RAWDEVICE /dev/rdsk/c13t22d0s1;
CREATE DBSPACE SP99  ALLOCATE
RAWDEVICE /dev/rdsk/c13t32d0s1;
CREATE DBSPACE SP100 ALLOCATE
RAWDEVICE /dev/rdsk/c21t38d0s1;
CREATE DBSPACE SP101 ALLOCATE
RAWDEVICE /dev/rdsk/c13t34d0s1;
CREATE DBSPACE SP102 ALLOCATE
RAWDEVICE /dev/rdsk/c13t35d0s1;
CREATE DBSPACE SP103 ALLOCATE
RAWDEVICE /dev/rdsk/c13t36d0s1;
CREATE DBSPACE SP104 ALLOCATE
RAWDEVICE /dev/rdsk/c13t37d0s1;
CREATE DBSPACE SP105 ALLOCATE
RAWDEVICE /dev/rdsk/c13t38d0s1;
CREATE DBSPACE SP106 ALLOCATE
RAWDEVICE /dev/rdsk/c13t48d0s1;
CREATE DBSPACE SP107 ALLOCATE
RAWDEVICE /dev/rdsk/c13t49d0s1;
CREATE DBSPACE SP108 ALLOCATE
RAWDEVICE /dev/rdsk/c13t50d0s1;
CREATE DBSPACE SP109 ALLOCATE
RAWDEVICE /dev/rdsk/c13t51d0s1;
CREATE DBSPACE SP110 ALLOCATE
RAWDEVICE /dev/rdsk/c13t52d0s1;
CREATE DBSPACE SP111 ALLOCATE
RAWDEVICE /dev/rdsk/c13t53d0s1;
CREATE DBSPACE SP112 ALLOCATE
RAWDEVICE /dev/rdsk/c13t54d0s1;
CREATE DBSPACE SP113 ALLOCATE
RAWDEVICE /dev/rdsk/c15t64d0s1;
CREATE DBSPACE SP114 ALLOCATE
RAWDEVICE /dev/rdsk/c15t65d0s1;
CREATE DBSPACE SP115 ALLOCATE
RAWDEVICE /dev/rdsk/c15t66d0s1;
CREATE DBSPACE SP116 ALLOCATE
RAWDEVICE /dev/rdsk/c15t67d0s1;
CREATE DBSPACE SP117 ALLOCATE
RAWDEVICE /dev/rdsk/c15t68d0s1;
CREATE DBSPACE SP118 ALLOCATE
RAWDEVICE /dev/rdsk/c15t69d0s1;
CREATE DBSPACE SP119 ALLOCATE
RAWDEVICE /dev/rdsk/c15t70d0s1;
CREATE DBSPACE SP120 ALLOCATE
RAWDEVICE /dev/rdsk/c15t80d0s1;
CREATE DBSPACE SP121 ALLOCATE
RAWDEVICE /dev/rdsk/c15t81d0s1;
CREATE DBSPACE SP122 ALLOCATE
RAWDEVICE /dev/rdsk/c15t82d0s1;
CREATE DBSPACE SP123 ALLOCATE
RAWDEVICE /dev/rdsk/c15t83d0s1;
CREATE DBSPACE SP124 ALLOCATE
RAWDEVICE /dev/rdsk/c15t84d0s1;
CREATE DBSPACE SP125 ALLOCATE
RAWDEVICE /dev/rdsk/c15t85d0s1;
CREATE DBSPACE SP126 ALLOCATE
RAWDEVICE /dev/rdsk/c15t86d0s1;
CREATE DBSPACE SP127 ALLOCATE
RAWDEVICE /dev/rdsk/c15t96d0s1;
CREATE DBSPACE SP128 ALLOCATE
RAWDEVICE /dev/rdsk/c15t97d0s1;
CREATE DBSPACE SP129 ALLOCATE
RAWDEVICE /dev/rdsk/c15t98d0s1;
CREATE DBSPACE SP130 ALLOCATE
RAWDEVICE /dev/rdsk/c15t99d0s1;
CREATE DBSPACE SP131 ALLOCATE
RAWDEVICE /dev/rdsk/c15t100d0s1;
CREATE DBSPACE SP132 ALLOCATE
RAWDEVICE /dev/rdsk/c15t101d0s1;
CREATE DBSPACE SP133 ALLOCATE
RAWDEVICE /dev/rdsk/c15t102d0s1;
CREATE DBSPACE SP134 ALLOCATE
RAWDEVICE /dev/rdsk/c15t112d0s1;
CREATE DBSPACE SP135 ALLOCATE
RAWDEVICE /dev/rdsk/c15t113d0s1;
CREATE DBSPACE SP136 ALLOCATE
RAWDEVICE /dev/rdsk/c15t114d0s1;
CREATE DBSPACE SP137 ALLOCATE
RAWDEVICE /dev/rdsk/c15t115d0s1;
CREATE DBSPACE SP138 ALLOCATE
RAWDEVICE /dev/rdsk/c15t116d0s1;
CREATE DBSPACE SP139 ALLOCATE
RAWDEVICE /dev/rdsk/c15t117d0s1;
CREATE DBSPACE SP140 ALLOCATE
RAWDEVICE /dev/rdsk/c15t118d0s1;
CREATE DBSPACE SP141 ALLOCATE
RAWDEVICE /dev/rdsk/c21t0d0s1;
CREATE DBSPACE SP142 ALLOCATE
RAWDEVICE /dev/rdsk/c21t1d0s1;
CREATE DBSPACE SP143 ALLOCATE
RAWDEVICE /dev/rdsk/c21t2d0s1;
CREATE DBSPACE SP144 ALLOCATE
RAWDEVICE /dev/rdsk/c21t3d0s1;
CREATE DBSPACE SP145 ALLOCATE
RAWDEVICE /dev/rdsk/c21t4d0s1;
CREATE DBSPACE SP146 ALLOCATE
RAWDEVICE /dev/rdsk/c21t5d0s1;
CREATE DBSPACE SP147 ALLOCATE
RAWDEVICE /dev/rdsk/c21t6d0s1;
CREATE DBSPACE SP148 ALLOCATE
RAWDEVICE /dev/rdsk/c21t16d0s1;
CREATE DBSPACE SP149 ALLOCATE
RAWDEVICE /dev/rdsk/c21t17d0s1;
CREATE DBSPACE SP150 ALLOCATE
RAWDEVICE /dev/rdsk/c21t18d0s1;
CREATE DBSPACE SP151 ALLOCATE
RAWDEVICE /dev/rdsk/c21t19d0s1;
CREATE DBSPACE SP152 ALLOCATE
RAWDEVICE /dev/rdsk/c21t20d0s1;
CREATE DBSPACE SP153 ALLOCATE
RAWDEVICE /dev/rdsk/c21t21d0s1;
CREATE DBSPACE SP154 ALLOCATE
RAWDEVICE /dev/rdsk/c21t22d0s1;
CREATE DBSPACE SP155 ALLOCATE
RAWDEVICE /dev/rdsk/c21t32d0s1;
CREATE DBSPACE SP156 ALLOCATE
RAWDEVICE /dev/rdsk/c21t33d0s1;
CREATE DBSPACE SP157 ALLOCATE
RAWDEVICE /dev/rdsk/c4t2d0s1;
CREATE DBSPACE SP158 ALLOCATE
RAWDEVICE /dev/rdsk/c4t3d0s1;
CREATE DBSPACE SP159 ALLOCATE
RAWDEVICE /dev/rdsk/c4t4d0s1;
CREATE DBSPACE SP160 ALLOCATE
RAWDEVICE /dev/rdsk/c20t2d0s1;
CREATE DBSPACE SP161 ALLOCATE
RAWDEVICE /dev/rdsk/c20t3d0s1;
CREATE DBSPACE SP162 ALLOCATE
RAWDEVICE /dev/rdsk/c20t4d0s1;
CREATE DBSPACE SP163 ALLOCATE
RAWDEVICE /dev/rdsk/c20t5d0s1;
CREATE DBSPACE SP164 ALLOCATE
RAWDEVICE /dev/rdsk/c20t9d0s1;
CREATE DBSPACE SP165 ALLOCATE
RAWDEVICE /dev/rdsk/c20t10d0s1;
CREATE DBSPACE SP166 ALLOCATE
RAWDEVICE /dev/rdsk/c20t11d0s1;
CREATE DBSPACE SP167 ALLOCATE
RAWDEVICE /dev/rdsk/c20t12d0s1;
CREATE DBSPACE SP168 ALLOCATE
RAWDEVICE /dev/rdsk/c20t13d0s1;

******** ddl_db.mak ********
--
/*====================================
==========================*/
--/*                                  */
--
/*====================================
==========================*/
 CREATE DATABASE TPCC;

******** 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 LOAD17_D = /rdb/loaddata/17
set LOAD18_D = /rdb/loaddata/18
set LOAD19_D = /rdb/loaddata/19
set LOAD20_D = /rdb/loaddata/20
set LOAD21_D = /rdb/loaddata/21
set LOAD22_D = /rdb/loaddata/22
set LOAD23_D = /rdb/loaddata/23
set LOAD24_D = /rdb/loaddata/24
set LOAD25_D = /rdb/loaddata/25
set LOAD26_D = /rdb/loaddata/26
set LOAD27_D = /rdb/loaddata/27
set LOAD28_D = /rdb/loaddata/28
set LOAD29_D = /rdb/loaddata/29
```

```
set LOAD30_D = /rdb/loaddata/30
set LOAD31_D = /rdb/loaddata/31
set LOAD32_D = /rdb/loaddata/32
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
set WK17_D = /rdb/sortwk17
set WK18_D = /rdb/sortwk18
set WK19_D = /rdb/sortwk19
set WK20_D = /rdb/sortwk20
set WK21_D = /rdb/sortwk21
set WK22_D = /rdb/sortwk22
set WK23_D = /rdb/sortwk23
set WK24_D = /rdb/sortwk24
set WK25_D = /rdb/sortwk25
set WK26_D = /rdb/sortwk26
set WK27_D = /rdb/sortwk27
set WK28_D = /rdb/sortwk28
set WK29_D = /rdb/sortwk29
set WK30_D = /rdb/sortwk30
set WK31_D = /rdb/sortwk31
set WK32_D = /rdb/sortwk32

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 \

                                                -
n $LOAD1_D/data
rm $LOAD1_D/data

## WAREHOUSE

foreach num ( 0 1 )

        @ make_s1 = $num * 960 + 1
        @ make_e1 = $num * 960 + 60
        @ make_s2 = $num * 960 + 61
        @ make_e2 = $num * 960 + 120
        @ make_s3 = $num * 960 + 121
        @ make_e3 = $num * 960 + 180
        @ make_s4 = $num * 960 + 181
        @ make_e4 = $num * 960 + 240
        @ make_s5 = $num * 960 + 241
        @ make_e5 = $num * 960 + 300
        @ make_s6 = $num * 960 + 301
        @ make_e6 = $num * 960 + 360
        @ make_s7 = $num * 960 + 361
        @ make_e7 = $num * 960 + 420
        @ make_s8 = $num * 960 + 421
        @ make_e8 = $num * 960 + 480
        @ make_s9 = $num * 960 + 481
        @ make_e9 = $num * 960 + 540
        @ make_s10 = $num * 960 + 541
        @ make_e10 = $num * 960 + 600
        @ make_s11 = $num * 960 + 601
        @ make_e11 = $num * 960 + 660
        @ make_s12 = $num * 960 + 661
        @ make_e12 = $num * 960 + 720
        @ make_s13 = $num * 960 + 721
        @ make_e13 = $num * 960 + 780
        @ make_s14 = $num * 960 + 781
        @ make_e14 = $num * 960 + 840
        @ make_s15 = $num * 960 + 841
        @ make_e15 = $num * 960 + 900
        @ make_s16 = $num * 960 + 901
        @ make_e16 = $num * 960 + 960

        @ dsi_num1 = $num * 16 + 1
        @ dsi_num2 = $num * 16 + 2
        @ dsi_num3 = $num * 16 + 3
        @ dsi_num4 = $num * 16 + 4
        @ dsi_num5 = $num * 16 + 5
        @ dsi_num6 = $num * 16 + 6
        @ dsi_num7 = $num * 16 + 7
        @ dsi_num8 = $num * 16 + 8
        @ dsi_num9 = $num * 16 + 9
        @ dsi_num10 = $num * 16 + 10
        @ dsi_num11 = $num * 16 + 11
        @ dsi_num12 = $num * 16 + 12
        @ dsi_num13 = $num * 16 + 13
        @ dsi_num14 = $num * 16 + 14
        @ dsi_num15 = $num * 16 + 15
        @ dsi_num16 = $num * 16 + 16
        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 &
        wttpccd1 $LOAD9_D $make_s9
$make_e9 W &
        wttpccd1 $LOAD10_D $make_s10
$make_e10 W &
        wttpccd1 $LOAD11_D $make_s11
$make_e11 W &
        wttpccd1 $LOAD12_D $make_s12
$make_e12 W &
        wttpccd1 $LOAD13_D $make_s13
$make_e13 W &
        wttpccd1 $LOAD14_D $make_s14
$make_e14 W &
        wttpccd1 $LOAD15_D $make_s15
$make_e15 W &
        wttpccd1 $LOAD16_D $make_s16
$make_e16 W &

        wait
        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 &
                timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num9\_DSI \
                                                -
s $WK17_D \
                                                -
s $WK18_D \
                                                -
n $LOAD9_D/WH$make_s9\_$make_e9 &
                timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num10\_DSI \
                                                -
s $WK19_D \
                                                -
s $WK20_D \
                                                -
n $LOAD10_D/WH$make_s10\_$make_e10 &
                timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num11\_DSI \
                                                -
s $WK21_D \
                                                -
s $WK22_D \
                                                -
n $LOAD11_D/WH$make_s11\_$make_e11 &
                timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num12\_DSI \
                                                -
s $WK23_D \
                                                -
s $WK24_D \
                                                -
n $LOAD12_D/WH$make_s12\_$make_e12 &
                timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num13\_DSI \
                                                -
s $WK25_D \
                                                -
s $WK26_D \
                                                -
n $LOAD13_D/WH$make_s13\_$make_e13 &
                timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num14\_DSI \
                                                -
s $WK27_D \
                                                -
s $WK28_D \
                                                -
n $LOAD14_D/WH$make_s14\_$make_e14 &
                timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num15\_DSI \
                                                -
s $WK29_D \
                                                -
s $WK30_D \
                                                -
n $LOAD15_D/WH$make_s15\_$make_e15 &
                timex rdbsloader -mi -i
$RDBDB.WAREHOUSE_$dsi_num16\_DSI \
                                                -
s $WK31_D \
                                                -
s $WK32_D \
                                                -
n $LOAD16_D/WH$make_s16\_$make_e16 &

                wait
rm /rdb/loaddata/*/WH*

end
## DISTRICT

foreach num ( 0 1 )

                @ make_s1 = $num * 960 + 1
                @ make_e1 = $num * 960 + 60
                @ make_s2 = $num * 960 + 61
                @ make_e2 = $num * 960 + 120
                @ make_s3 = $num * 960 + 121
                @ make_e3 = $num * 960 + 180
                @ make_s4 = $num * 960 + 181
                @ make_e4 = $num * 960 + 240
                @ make_s5 = $num * 960 + 241
                @ make_e5 = $num * 960 + 300
                @ make_s6 = $num * 960 + 301
                @ make_e6 = $num * 960 + 360
                @ make_s7 = $num * 960 + 361
                @ make_e7 = $num * 960 + 420
                @ make_s8 = $num * 960 + 421
                @ make_e8 = $num * 960 + 480
                @ make_s9 = $num * 960 + 481
                @ make_e9 = $num * 960 + 540
                @ make_s10 = $num * 960 + 541
                @ make_e10 = $num * 960 + 600
                @ make_s11 = $num * 960 + 601
                @ make_e11 = $num * 960 + 660
                @ make_s12 = $num * 960 + 661
                @ make_e12 = $num * 960 + 720
                @ make_s13 = $num * 960 + 721
                @ make_e13 = $num * 960 + 780
                @ make_s14 = $num * 960 + 781
                @ make_e14 = $num * 960 + 840
                @ make_s15 = $num * 960 + 841
                @ make_e15 = $num * 960 + 900
                @ make_s16 = $num * 960 + 901
                @ make_e16 = $num * 960 + 960

                @ dsi_num1 = $num * 16 + 1
                @ dsi_num2 = $num * 16 + 2
                @ dsi_num3 = $num * 16 + 3
                @ dsi_num4 = $num * 16 + 4
                @ dsi_num5 = $num * 16 + 5
                @ dsi_num6 = $num * 16 + 6
                @ dsi_num7 = $num * 16 + 7
                @ dsi_num8 = $num * 16 + 8
                @ dsi_num9 = $num * 16 + 9
                @ dsi_num10 = $num * 16 + 10
                @ dsi_num11 = $num * 16 + 11
                @ dsi_num12 = $num * 16 + 12
                @ dsi_num13 = $num * 16 + 13
                @ dsi_num14 = $num * 16 + 14
                @ dsi_num15 = $num * 16 + 15
                @ dsi_num16 = $num * 16 + 16
                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 &
                wttpccd1 $LOAD9_D $make_s9
$make_e9 D &
                wttpccd1 $LOAD10_D $make_s10
$make_e10 D &
                wttpccd1 $LOAD11_D $make_s11
$make_e11 D &
                wttpccd1 $LOAD12_D $make_s12
$make_e12 D &
                wttpccd1 $LOAD13_D $make_s13
$make_e13 D &
                wttpccd1 $LOAD14_D $make_s14
$make_e14 D &
                wttpccd1 $LOAD15_D $make_s15
$make_e15 D &
                wttpccd1 $LOAD16_D $make_s16
$make_e16 D &

                wait
                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 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num9\_DSI \
-
s $WK17_D \
-
s $WK18_D \
-
n $LOAD9_D/DI$make_s9\_$make_e9 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num10\_DSI \
-
s $WK19_D \
-
s $WK20_D \
-
n $LOAD10_D/DI$make_s10\_$make_e10 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num11\_DSI \
-
s $WK21_D \
-
s $WK22_D \
-
n $LOAD11_D/DI$make_s11\_$make_e11 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num12\_DSI \
-
s $WK23_D \
-
s $WK24_D \
-
n $LOAD12_D/DI$make_s12\_$make_e12 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num13\_DSI \
-
s $WK25_D \
-
s $WK26_D \
-
n $LOAD13_D/DI$make_s13\_$make_e13 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num14\_DSI \
-
s $WK27_D \
-
s $WK28_D \
-
n $LOAD14_D/DI$make_s14\_$make_e14 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num15\_DSI \
-
s $WK29_D \
-
s $WK30_D \
-
n $LOAD15_D/DI$make_s15\_$make_e15 &
        timex rdbsloader -mi -i
$RDBDB.DISTRICT_$dsi_num16\_DSI \
-
s $WK31_D \
-
s $WK32_D \
-
n $LOAD16_D/DI$make_s16\_$make_e16 &

        wait
rm /rdb/loaddata/*/DI*

end
## CUSTOMER

foreach num ( 0 1 2 3 4 5 6 7 8 9 10 )

        @ make_s1 = $num * 160 + 1
        @ make_e1 = $num * 160 + 10
        @ make_s2 = $num * 160 + 11
        @ make_e2 = $num * 160 + 20
        @ make_s3 = $num * 160 + 21
        @ make_e3 = $num * 160 + 30
        @ make_s4 = $num * 160 + 31
        @ make_e4 = $num * 160 + 40
        @ make_s5 = $num * 160 + 41
        @ make_e5 = $num * 160 + 50
        @ make_s6 = $num * 160 + 51
        @ make_e6 = $num * 160 + 60
        @ make_s7 = $num * 160 + 61
        @ make_e7 = $num * 160 + 70
        @ make_s8 = $num * 160 + 71
        @ make_e8 = $num * 160 + 80
        @ make_s9 = $num * 160 + 81
        @ make_e9 = $num * 160 + 90
        @ make_s10 = $num * 160 + 91
        @ make_e10 = $num * 160 + 100
        @ make_s11 = $num * 160 + 101
        @ make_e11 = $num * 160 + 110
        @ make_s12 = $num * 160 + 111
        @ make_e12 = $num * 160 + 120
```

```
        @ make_s13 = $num * 160 + 121
        @ make_e13 = $num * 160 + 130
        @ make_s14 = $num * 160 + 131
        @ make_e14 = $num * 160 + 140
        @ make_s15 = $num * 160 + 141
        @ make_e15 = $num * 160 + 150
        @ make_s16 = $num * 160 + 151
        @ make_e16 = $num * 160 + 160

        @ dsi_num1 = $num * 16 + 1
        @ dsi_num2 = $num * 16 + 2
        @ dsi_num3 = $num * 16 + 3
        @ dsi_num4 = $num * 16 + 4
        @ dsi_num5 = $num * 16 + 5
        @ dsi_num6 = $num * 16 + 6
        @ dsi_num7 = $num * 16 + 7
        @ dsi_num8 = $num * 16 + 8
        @ dsi_num9 = $num * 16 + 9
        @ dsi_num10 = $num * 16 + 10
        @ dsi_num11 = $num * 16 + 11
        @ dsi_num12 = $num * 16 + 12
        @ dsi_num13 = $num * 16 + 13
        @ dsi_num14 = $num * 16 + 14
        @ dsi_num15 = $num * 16 + 15
        @ dsi_num16 = $num * 16 + 16
        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 &
        wttpccd1 $LOAD9_D $make_s9
$make_e9 C &
        wttpccd1 $LOAD10_D $make_s10
$make_e10 C &
        wttpccd1 $LOAD11_D $make_s11
$make_e11 C &
        wttpccd1 $LOAD12_D $make_s12
$make_e12 C &
        wttpccd1 $LOAD13_D $make_s13
$make_e13 C &
        wttpccd1 $LOAD14_D $make_s14
$make_e14 C &
        wttpccd1 $LOAD15_D $make_s15
$make_e15 C &
        wttpccd1 $LOAD16_D $make_s16
$make_e16 C &

        wait
        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 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num9\_DSI -h \
                                    -
s $WK17_D \
                                    -
s $WK18_D \
                                    -
n $LOAD9_D/CU$make_s9\_$make_e9 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num10\_DSI -h \
                                    -
s $WK19_D \
                                    -
s $WK20_D \
                                    -
n $LOAD10_D/CU$make_s10\_$make_e10 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num11\_DSI -h \
                                    -
s $WK21_D \
                                    -
s $WK22_D \
                                    -
n $LOAD11_D/CU$make_s11\_$make_e11 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num12\_DSI -h \
                                    -
s $WK23_D \
                                    -
s $WK24_D \
                                    -
n $LOAD12_D/CU$make_s12\_$make_e12 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num13\_DSI -h \
                                    -
s $WK25_D \
                                    -
s $WK26_D \
                                    -
n $LOAD13_D/CU$make_s13\_$make_e13 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num14\_DSI -h \
```

```
                                                -
s $WK27_D \

                                                -
s $WK28_D \

                                                -
n $LOAD14_D/CU$make_s14\_$make_e14 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num15\_DSI -h \

                                                -
s $WK29_D \

                                                -
s $WK30_D \

                                                -
n $LOAD15_D/CU$make_s15\_$make_e15 &
        timex rdbsloader -mi -i
$RDBDB.CUSTOMER_$dsi_num16\_DSI -h \

                                                -
s $WK31_D \

                                                -
s $WK32_D \

                                                -
n $LOAD16_D/CU$make_s16\_$make_e16 &

        wait
rm /rdb/loaddata/*/CU*

end
## HISTORY

foreach num ( 0 1 2 3 4 5 6 7 8 9 10 )

        @ make_s1 = $num * 160 + 1
        @ make_e1 = $num * 160 + 10
        @ make_s2 = $num * 160 + 11
        @ make_e2 = $num * 160 + 20
        @ make_s3 = $num * 160 + 21
        @ make_e3 = $num * 160 + 30
        @ make_s4 = $num * 160 + 31
        @ make_e4 = $num * 160 + 40
        @ make_s5 = $num * 160 + 41
        @ make_e5 = $num * 160 + 50
        @ make_s6 = $num * 160 + 51
        @ make_e6 = $num * 160 + 60
        @ make_s7 = $num * 160 + 61
        @ make_e7 = $num * 160 + 70
        @ make_s8 = $num * 160 + 71
        @ make_e8 = $num * 160 + 80
        @ make_s9 = $num * 160 + 81
        @ make_e9 = $num * 160 + 90
        @ make_s10 = $num * 160 + 91
        @ make_e10 = $num * 160 + 100
        @ make_s11 = $num * 160 + 101
        @ make_e11 = $num * 160 + 110
        @ make_s12 = $num * 160 + 111
        @ make_e12 = $num * 160 + 120
        @ make_s13 = $num * 160 + 121
        @ make_e13 = $num * 160 + 130
        @ make_s14 = $num * 160 + 131
        @ make_e14 = $num * 160 + 140
        @ make_s15 = $num * 160 + 141
        @ make_e15 = $num * 160 + 150
        @ make_s16 = $num * 160 + 151
        @ make_e16 = $num * 160 + 160

        @ dsi_num1 = $num * 16 + 1
        @ dsi_num2 = $num * 16 + 2
        @ dsi_num3 = $num * 16 + 3
        @ dsi_num4 = $num * 16 + 4
        @ dsi_num5 = $num * 16 + 5
        @ dsi_num6 = $num * 16 + 6
        @ dsi_num7 = $num * 16 + 7
        @ dsi_num8 = $num * 16 + 8
        @ dsi_num9 = $num * 16 + 9
        @ dsi_num10 = $num * 16 + 10
        @ dsi_num11 = $num * 16 + 11
        @ dsi_num12 = $num * 16 + 12
        @ dsi_num13 = $num * 16 + 13
        @ dsi_num14 = $num * 16 + 14
        @ dsi_num15 = $num * 16 + 15
        @ dsi_num16 = $num * 16 + 16
        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 &
        wttpccd1 $LOAD9_D $make_s9
$make_e9 H &
        wttpccd1 $LOAD10_D $make_s10
$make_e10 H &
        wttpccd1 $LOAD11_D $make_s11
$make_e11 H &
        wttpccd1 $LOAD12_D $make_s12
$make_e12 H &
        wttpccd1 $LOAD13_D $make_s13
$make_e13 H &
        wttpccd1 $LOAD14_D $make_s14
$make_e14 H &
        wttpccd1 $LOAD15_D $make_s15
$make_e15 H &
        wttpccd1 $LOAD16_D $make_s16
$make_e16 H &

        wait
        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 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num9\_DSI \

-
s $WK17_D \

-
s $WK18_D \

-
n $LOAD9_D/HI$make_s9\_$make_e9 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num10\_DSI \

-
s $WK19_D \

-
s $WK20_D \

-
n $LOAD10_D/HI$make_s10\_$make_e10 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num11\_DSI \

-
s $WK21_D \

-
s $WK22_D \

-
n $LOAD11_D/HI$make_s11\_$make_e11 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num12\_DSI \

-
s $WK23_D \

-
s $WK24_D \

-
n $LOAD12_D/HI$make_s12\_$make_e12 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num13\_DSI \

-
s $WK25_D \

-
s $WK26_D \

-
n $LOAD13_D/HI$make_s13\_$make_e13 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num14\_DSI \

-
s $WK27_D \

-
s $WK28_D \

-
n $LOAD14_D/HI$make_s14\_$make_e14 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num15\_DSI \

-
s $WK29_D \

-
s $WK30_D \

-
n $LOAD15_D/HI$make_s15\_$make_e15 &
        timex rdbsloader -mi -i
$RDBDB.HISTORY_$dsi_num16\_DSI \

-
s $WK31_D \

-
s $WK32_D \

-
n $LOAD16_D/HI$make_s16\_$make_e16 &

        wait
rm /rdb/loaddata/*/HI*

end
## STOCK

foreach num ( 0 1 2 3 )

        @ make_s1 = $num * 480 + 1
        @ make_e1 = $num * 480 + 30
        @ make_s2 = $num * 480 + 31
        @ make_e2 = $num * 480 + 60
        @ make_s3 = $num * 480 + 61
        @ make_e3 = $num * 480 + 90
        @ make_s4 = $num * 480 + 91
        @ make_e4 = $num * 480 + 120
        @ make_s5 = $num * 480 + 121
        @ make_e5 = $num * 480 + 150
        @ make_s6 = $num * 480 + 151
        @ make_e6 = $num * 480 + 180
        @ make_s7 = $num * 480 + 181
        @ make_e7 = $num * 480 + 210
        @ make_s8 = $num * 480 + 211
        @ make_e8 = $num * 480 + 240
        @ make_s9 = $num * 480 + 241
        @ make_e9 = $num * 480 + 270
        @ make_s10 = $num * 480 + 271
        @ make_e10 = $num * 480 + 300
        @ make_s11 = $num * 480 + 301
        @ make_e11 = $num * 480 + 330
        @ make_s12 = $num * 480 + 331
        @ make_e12 = $num * 480 + 360
        @ make_s13 = $num * 480 + 361
        @ make_e13 = $num * 480 + 390
        @ make_s14 = $num * 480 + 391
        @ make_e14 = $num * 480 + 420
        @ make_s15 = $num * 480 + 421
        @ make_e15 = $num * 480 + 450
        @ make_s16 = $num * 480 + 451
        @ make_e16 = $num * 480 + 480

        @ dsi_num1 = $num * 16 + 1
        @ dsi_num2 = $num * 16 + 2
        @ dsi_num3 = $num * 16 + 3
        @ dsi_num4 = $num * 16 + 4
        @ dsi_num5 = $num * 16 + 5
        @ dsi_num6 = $num * 16 + 6
        @ dsi_num7 = $num * 16 + 7
        @ dsi_num8 = $num * 16 + 8
        @ dsi_num9 = $num * 16 + 9
        @ dsi_num10 = $num * 16 + 10
        @ dsi_num11 = $num * 16 + 11
        @ dsi_num12 = $num * 16 + 12
        @ dsi_num13 = $num * 16 + 13
        @ dsi_num14 = $num * 16 + 14
        @ dsi_num15 = $num * 16 + 15
        @ dsi_num16 = $num * 16 + 16
        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_s9
$make_e9 S &
        wttpccd1 $LOAD10_D $make_s10
$make_e10 S &
        wttpccd1 $LOAD11_D $make_s11
$make_e11 S &
        wttpccd1 $LOAD12_D $make_s12
$make_e12 S &
        wttpccd1 $LOAD13_D $make_s13
$make_e13 S &
        wttpccd1 $LOAD14_D $make_s14
$make_e14 S &
        wttpccd1 $LOAD15_D $make_s15
$make_e15 S &
        wttpccd1 $LOAD16_D $make_s16
$make_e16 S &

        wait
        timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num1\_DSI \

                                        -
s $WK1_D \

                                        -
s $WK2_D \

                                        -
n $LOAD1_D/ST$make_s1\_$make_e1 &
        timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num2\_DSI \

                                        -
s $WK3_D \

                                        -
s $WK4_D \
```

```
                                                -
n $LOAD2_D/ST$make_s2\_$make_e2 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num3\_DSI \
                                                -
s $WK5_D \
                                                -
s $WK6_D \
                                                -
n $LOAD3_D/ST$make_s3\_$make_e3 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num4\_DSI \
                                                -
s $WK7_D \
                                                -
s $WK8_D \
                                                -
n $LOAD4_D/ST$make_s4\_$make_e4 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num5\_DSI \
                                                -
s $WK9_D \
                                                -
s $WK10_D \
                                                -
n $LOAD5_D/ST$make_s5\_$make_e5 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num6\_DSI \
                                                -
s $WK11_D \
                                                -
s $WK12_D \
                                                -
n $LOAD6_D/ST$make_s6\_$make_e6 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num7\_DSI \
                                                -
s $WK13_D \
                                                -
s $WK14_D \
                                                -
n $LOAD7_D/ST$make_s7\_$make_e7 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num8\_DSI \
                                                -
s $WK15_D \
                                                -
s $WK16_D \
                                                -
n $LOAD8_D/ST$make_s8\_$make_e8 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num9\_DSI \
                                                -
s $WK17_D \
                                                -
s $WK18_D \
                                                -
n $LOAD9_D/ST$make_s9\_$make_e9 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num10\_DSI \
                                                -
s $WK19_D \
                                                -
s $WK20_D \
                                                -
n $LOAD10_D/ST$make_s10\_$make_e10 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num11\_DSI \
                                                -
s $WK21_D \
                                                -
s $WK22_D \
                                                -
n $LOAD11_D/ST$make_s11\_$make_e11 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num12\_DSI \
                                                -
s $WK23_D \
                                                -
s $WK24_D \
                                                -
n $LOAD12_D/ST$make_s12\_$make_e12 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num13\_DSI \
                                                -
s $WK25_D \
                                                -
s $WK26_D \
                                                -
n $LOAD13_D/ST$make_s13\_$make_e13 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num14\_DSI \
                                                -
s $WK27_D \
                                                -
s $WK28_D \
                                                -
n $LOAD14_D/ST$make_s14\_$make_e14 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num15\_DSI \
                                                -
s $WK29_D \
                                                -
s $WK30_D \
                                                -
n $LOAD15_D/ST$make_s15\_$make_e15 &
                timex rdbsloader -mi -i
$RDBDB.STOCK_$dsi_num16\_DSI \
                                                -
s $WK31_D \
                                                -
s $WK32_D \
                                                -
n $LOAD16_D/ST$make_s16\_$make_e16 &

                wait
rm /rdb/loaddata/*/ST*

end
## ORDERS ORDERLINE NEWORDER

foreach num ( 0 1 2 3 4 5 6 7 8 9 10 )

                @ make_s1 = $num * 160 + 1
                @ make_e1 = $num * 160 + 10
                @ make_s2 = $num * 160 + 11
                @ make_e2 = $num * 160 + 20
                @ make_s3 = $num * 160 + 21
                @ make_e3 = $num * 160 + 30
                @ make_s4 = $num * 160 + 31
                @ make_e4 = $num * 160 + 40
                @ make_s5 = $num * 160 + 41
                @ make_e5 = $num * 160 + 50
                @ make_s6 = $num * 160 + 51
                @ make_e6 = $num * 160 + 60
                @ make_s7 = $num * 160 + 61
                @ make_e7 = $num * 160 + 70
                @ make_s8 = $num * 160 + 71
                @ make_e8 = $num * 160 + 80
                @ make_s9 = $num * 160 + 81
                @ make_e9 = $num * 160 + 90
                @ make_s10 = $num * 160 + 91
                @ make_e10 = $num * 160 + 100
                @ make_s11 = $num * 160 + 101
                @ make_e11 = $num * 160 + 110
                @ make_s12 = $num * 160 + 111
                @ make_e12 = $num * 160 + 120
                @ make_s13 = $num * 160 + 121
                @ make_e13 = $num * 160 + 130
                @ make_s14 = $num * 160 + 131
                @ make_e14 = $num * 160 + 140
                @ make_s15 = $num * 160 + 141
                @ make_e15 = $num * 160 + 150
                @ make_s16 = $num * 160 + 151
                @ make_e16 = $num * 160 + 160

                @ dsi_num_os1 = $num * 16 * 1 + 1
                @ dsi_num_os2 = $num * 16 * 1 + 2
                @ dsi_num_os3 = $num * 16 * 1 + 3
                @ dsi_num_os4 = $num * 16 * 1 + 4
                @ dsi_num_os5 = $num * 16 * 1 + 5
                @ dsi_num_os6 = $num * 16 * 1 + 6
                @ dsi_num_os7 = $num * 16 * 1 + 7
```

```
        @ dsi_num_os8 = $num * 16 * 1 + 8
        @ dsi_num_os9 = $num * 16 * 1 + 9
        @ dsi_num_os10 = $num * 16 * 1 +
10
        @ dsi_num_os11 = $num * 16 * 1 +
11
        @ dsi_num_os12 = $num * 16 * 1 +
12
        @ dsi_num_os13 = $num * 16 * 1 +
13
        @ dsi_num_os14 = $num * 16 * 1 +
14
        @ dsi_num_os15 = $num * 16 * 1 +
15
        @ dsi_num_os16 = $num * 16 * 1 +
16
        @ dsi_num_ol1 = $num * 16 * 1 + 1
        @ dsi_num_ol2 = $num * 16 * 1 + 2
        @ dsi_num_ol3 = $num * 16 * 1 + 3
        @ dsi_num_ol4 = $num * 16 * 1 + 4
        @ dsi_num_ol5 = $num * 16 * 1 + 5
        @ dsi_num_ol6 = $num * 16 * 1 + 6
        @ dsi_num_ol7 = $num * 16 * 1 + 7
        @ dsi_num_ol8 = $num * 16 * 1 + 8
        @ dsi_num_ol9 = $num * 16 * 1 + 9
        @ dsi_num_ol10 = $num * 16 * 1 +
10
        @ dsi_num_ol11 = $num * 16 * 1 +
11
        @ dsi_num_ol12 = $num * 16 * 1 +
12
        @ dsi_num_ol13 = $num * 16 * 1 +
13
        @ dsi_num_ol14 = $num * 16 * 1 +
14
        @ dsi_num_ol15 = $num * 16 * 1 +
15
        @ dsi_num_ol16 = $num * 16 * 1 +
16
        @ dsi_num_no1 = $num * 16 * 1 + 1
        @ dsi_num_no2 = $num * 16 * 1 + 2
        @ dsi_num_no3 = $num * 16 * 1 + 3
        @ dsi_num_no4 = $num * 16 * 1 + 4
        @ dsi_num_no5 = $num * 16 * 1 + 5
        @ dsi_num_no6 = $num * 16 * 1 + 6
        @ dsi_num_no7 = $num * 16 * 1 + 7
        @ dsi_num_no8 = $num * 16 * 1 + 8
        @ dsi_num_no9 = $num * 16 * 1 + 9
        @ dsi_num_no10 = $num * 16 * 1 +
10
        @ dsi_num_no11 = $num * 16 * 1 +
11
        @ dsi_num_no12 = $num * 16 * 1 +
12
        @ dsi_num_no13 = $num * 16 * 1 +
13
        @ dsi_num_no14 = $num * 16 * 1 +
14
        @ dsi_num_no15 = $num * 16 * 1 +
15
        @ dsi_num_no16 = $num * 16 * 1 +
16
        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 &
        wttpccd1 $LOAD9_D $make_s9
$make_e9 O &
        wttpccd1 $LOAD10_D $make_s10
$make_e10 O &
        wttpccd1 $LOAD11_D $make_s11
$make_e11 O &
        wttpccd1 $LOAD12_D $make_s12
$make_e12 O &
        wttpccd1 $LOAD13_D $make_s13
$make_e13 O &
        wttpccd1 $LOAD14_D $make_s14
$make_e14 O &
        wttpccd1 $LOAD15_D $make_s15
$make_e15 O &
        wttpccd1 $LOAD16_D $make_s16
$make_e16 O &

        wait
        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_e
1 &
        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_e
2 &
        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_e
3 &
        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_e
4 &
        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_e
5 &
        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_e
6 &
        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_e
7 &
        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_e
8 &
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os9\_DSI -h -f 10\
                                                -
s $WK17_D \

                                                -
s $WK18_D \

        $LOAD9_D/OS$make_s9\_$make_e
9 &
```

```
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os10\_DSI -h -f
10\
                                                -
s $WK19_D \
                                                -
s $WK20_D \
        $LOAD10_D/OS$make_s10\_$make
_e10 &
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os11\_DSI -h -f
10\
                                                -
s $WK21_D \
                                                -
s $WK22_D \
        $LOAD11_D/OS$make_s11\_$make
_e11 &
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os12\_DSI -h -f
10\
                                                -
s $WK23_D \
                                                -
s $WK24_D \
        $LOAD12_D/OS$make_s12\_$make
_e12 &
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os13\_DSI -h -f
10\
                                                -
s $WK25_D \
                                                -
s $WK26_D \
        $LOAD13_D/OS$make_s13\_$make
_e13 &
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os14\_DSI -h -f
10\
                                                -
s $WK27_D \
                                                -
s $WK28_D \
        $LOAD14_D/OS$make_s14\_$make
_e14 &
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os15\_DSI -h -f
10\
                                                -
s $WK29_D \
                                                -
s $WK30_D \
        $LOAD15_D/OS$make_s15\_$make
_e15 &
        timex rdbsloader -mi -i
$RDBDB.ORDERS_$dsi_num_os16\_DSI -h -f
10\
                                                -
s $WK31_D \
                                                -
s $WK32_D \
        $LOAD16_D/OS$make_s16\_$make
_e16 &
        wait
        @ dsi_num_os1 = $dsi_num_os1 +
16
        @ dsi_num_os2 = $dsi_num_os2 +
16
        @ dsi_num_os3 = $dsi_num_os3 +
16
        @ dsi_num_os4 = $dsi_num_os4 +
16
        @ dsi_num_os5 = $dsi_num_os5 +
16
        @ dsi_num_os6 = $dsi_num_os6 +
16
        @ dsi_num_os7 = $dsi_num_os7 +
16
        @ dsi_num_os8 = $dsi_num_os8 +
16
        @ dsi_num_os9 = $dsi_num_os9 +
16
        @ dsi_num_os10 = $dsi_num_os10
+ 16
        @ dsi_num_os11 = $dsi_num_os11
+ 16
        @ dsi_num_os12 = $dsi_num_os12
+ 16
        @ dsi_num_os13 = $dsi_num_os13
+ 16
        @ dsi_num_os14 = $dsi_num_os14
+ 16
        @ dsi_num_os15 = $dsi_num_os15
+ 16
        @ dsi_num_os16 = $dsi_num_os16
+ 16
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol1\_DSI -h \
                                                -
s $WK1_D \
                                                -
s $WK2_D \
        $LOAD1_D/OL$make_s1\_$make_e
1 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol2\_DSI -h \
                                                -
s $WK3_D \
                                                -
s $WK4_D \
        $LOAD2_D/OL$make_s2\_$make_e
2 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol3\_DSI -h \
                                                -
s $WK5_D \
                                                -
s $WK6_D \
        $LOAD3_D/OL$make_s3\_$make_e
3 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol4\_DSI -h \
                                                -
s $WK7_D \
                                                -
s $WK8_D \
        $LOAD4_D/OL$make_s4\_$make_e
4 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol5\_DSI -h \
                                                -
s $WK9_D \
                                                -
s $WK10_D \
        $LOAD5_D/OL$make_s5\_$make_e
5 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol6\_DSI -h \
                                                -
s $WK11_D \
                                                -
s $WK12_D \
        $LOAD6_D/OL$make_s6\_$make_e
6 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol7\_DSI -h \
                                                -
s $WK13_D \
                                                -
s $WK14_D \
```

```
        $LOAD7_D/OL$make_s7\_$make_e
7 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol8\_DSI -h \
                                                -
s $WK15_D \
                                                -
s $WK16_D \

        $LOAD8_D/OL$make_s8\_$make_e
8 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol9\_DSI -h \
                                                -
s $WK17_D \
                                                -
s $WK18_D \

        $LOAD9_D/OL$make_s9\_$make_e
9 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol10\_DSI -h \
                                                -
s $WK19_D \
                                                -
s $WK20_D \

        $LOAD10_D/OL$make_s10\_$make
_e10 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol11\_DSI -h \
                                                -
s $WK21_D \
                                                -
s $WK22_D \

        $LOAD11_D/OL$make_s11\_$make
_e11 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol12\_DSI -h \
                                                -
s $WK23_D \
                                                -
s $WK24_D \

        $LOAD12_D/OL$make_s12\_$make
_e12 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol13\_DSI -h \
                                                -
s $WK25_D \
                                                -
s $WK26_D \

        $LOAD13_D/OL$make_s13\_$make
_e13 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol14\_DSI -h \
                                                -
s $WK27_D \
                                                -
s $WK28_D \

        $LOAD14_D/OL$make_s14\_$make
_e14 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol15\_DSI -h \
                                                -
s $WK29_D \
                                                -
s $WK30_D \

        $LOAD15_D/OL$make_s15\_$make
_e15 &
        timex rdbsloader -mi -i
$RDBDB.ORDERLIN_$dsi_num_ol16\_DSI -h \
                                                -
s $WK31_D \
                                                -
s $WK32_D \

        $LOAD16_D/OL$make_s16\_$make
_e16 &

        wait
        @ dsi_num_ol1 = $dsi_num_ol1 + 16
        @ dsi_num_ol2 = $dsi_num_ol2 + 16
        @ dsi_num_ol3 = $dsi_num_ol3 + 16
        @ dsi_num_ol4 = $dsi_num_ol4 + 16
        @ dsi_num_ol5 = $dsi_num_ol5 + 16
        @ dsi_num_ol6 = $dsi_num_ol6 + 16
        @ dsi_num_ol7 = $dsi_num_ol7 + 16
        @ dsi_num_ol8 = $dsi_num_ol8 + 16
        @ dsi_num_ol9 = $dsi_num_ol9 + 16
        @ dsi_num_ol10 = $dsi_num_ol10 +
16
        @ dsi_num_ol11 = $dsi_num_ol11 +
16
        @ dsi_num_ol12 = $dsi_num_ol12 +
16
        @ dsi_num_ol13 = $dsi_num_ol13 +
16
        @ dsi_num_ol14 = $dsi_num_ol14 +
16
        @ dsi_num_ol15 = $dsi_num_ol15 +
16
        @ dsi_num_ol16 = $dsi_num_ol16 +
16
        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 &
        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 &
        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 &
        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 &
        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 &
        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 &
                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 &
                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 &
                timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no9\_DSI -h -f
20 \

                                                -
s $WK17_D \

                                                -
s $WK18_D \

                                                -
n $LOAD9_D/NO$make_s9\_$make_e9 &
                timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no10\_DSI -h -
f 20 \

                                                -
s $WK19_D \

                                                -
s $WK20_D \

                                                -
n $LOAD10_D/NO$make_s10\_$make_e10 &
                timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no11\_DSI -h -
f 20 \

                                                -
s $WK21_D \

                                                -
s $WK22_D \

                                                -
n $LOAD11_D/NO$make_s11\_$make_e11 &
                timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no12\_DSI -h -
f 20 \

                                                -
s $WK23_D \

                                                -
s $WK24_D \

                                                -
n $LOAD12_D/NO$make_s12\_$make_e12 &
                timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no13\_DSI -h -
f 20 \

                                                -
s $WK25_D \

                                                -
s $WK26_D \

                                                -
n $LOAD13_D/NO$make_s13\_$make_e13 &
                timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no14\_DSI -h -
f 20 \

                                                -
s $WK27_D \

                                                -
s $WK28_D \

                                                -
n $LOAD14_D/NO$make_s14\_$make_e14 &
                timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no15\_DSI -h -
f 20 \

                                                -
s $WK29_D \

                                                -
s $WK30_D \

                                                -
n $LOAD15_D/NO$make_s15\_$make_e15 &
                timex rdbsloader -mi -i
$RDBDB.NEWORDER_$dsi_num_no16\_DSI -h -
f 20 \

                                                -
s $WK31_D \

                                                -
s $WK32_D \

                                                -
n $LOAD16_D/NO$make_s16\_$make_e16 &

                wait
                @ dsi_num_no1 = $dsi_num_no1 +
16
                @ dsi_num_no2 = $dsi_num_no2 +
16
                @ dsi_num_no3 = $dsi_num_no3 +
16
                @ dsi_num_no4 = $dsi_num_no4 +
16
                @ dsi_num_no5 = $dsi_num_no5 +
16
                @ dsi_num_no6 = $dsi_num_no6 +
16
                @ dsi_num_no7 = $dsi_num_no7 +
16
                @ dsi_num_no8 = $dsi_num_no8 +
16
                @ dsi_num_no9 = $dsi_num_no9 +
16
                @ dsi_num_no10 = $dsi_num_no10
+ 16
                @ dsi_num_no11 = $dsi_num_no11
+ 16
                @ dsi_num_no12 = $dsi_num_no12
+ 16
                @ dsi_num_no13 = $dsi_num_no13
+ 16
                @ dsi_num_no14 = $dsi_num_no14
+ 16
                @ dsi_num_no15 = $dsi_num_no15
+ 16
                @ dsi_num_no16 = $dsi_num_no16
+ 16
rm /rdb/loaddata/*/OS*

rm /rdb/loaddata/*/OL*

rm /rdb/loaddata/*/NO*

end
******** mkarc.sh ********
:
set -x


timex rdblog -G -a /dev/rdsk/c2t67d0s1 1725M
sleep 1
timex rdblog -U -a /dev/rdsk/c2t67d0s3
sleep 1
timex rdblog -U -a /dev/rdsk/c2t67d0s4
sleep 1
timex rdblog -U -a /dev/rdsk/c2t67d0s5
sleep 1
timex rdblog -U -a /dev/rdsk/c2t67d0s6
sleep 1
timex rdblog -U -a /dev/rdsk/c12t33d0s1
sleep 1
timex rdblog -U -a /dev/rdsk/c12t33d0s3
sleep 1
timex rdblog -U -a /dev/rdsk/c12t33d0s4
sleep 1
timex rdblog -U -a /dev/rdsk/c12t33d0s5
sleep 1
timex rdblog -U -a /dev/rdsk/c12t33d0s6
sleep 1
timex rdblog -U -a /dev/rdsk/c21t34d0s1
sleep 1
timex rdblog -U -a /dev/rdsk/c21t34d0s3
sleep 1
timex rdblog -U -a /dev/rdsk/c21t34d0s4
sleep 1
timex rdblog -U -a /dev/rdsk/c21t34d0s5
sleep 1
timex rdblog -U -a /dev/rdsk/c21t34d0s6
sleep 1
timex rdblog -U -a /dev/rdsk/c21t35d0s1
sleep 1
timex rdblog -U -a /dev/rdsk/c21t35d0s3
sleep 1
timex rdblog -U -a /dev/rdsk/c21t35d0s4
sleep 1
timex rdblog -U -a /dev/rdsk/c21t35d0s5
```

```
sleep 1
timex rdblog -U -a /dev/rdsk/c21t35d0s6
sleep 1
timex rdblog -U -a /dev/rdsk/c21t36d0s1
sleep 1
timex rdblog -U -a /dev/rdsk/c21t36d0s3
sleep 1
timex rdblog -U -a /dev/rdsk/c21t36d0s4
sleep 1
timex rdblog -U -a /dev/rdsk/c21t36d0s5
sleep 1
timex rdblog -U -a /dev/rdsk/c21t36d0s6
sleep 1
timex rdblog -U -a /dev/rdsk/c21t37d0s1
sleep 1
timex rdblog -U -a /dev/rdsk/c21t37d0s3
sleep 1
timex rdblog -U -a /dev/rdsk/c21t37d0s4
sleep 1
timex rdblog -U -a /dev/rdsk/c21t37d0s5
sleep 1
timex rdblog -U -a /dev/rdsk/c21t37d0s6
sleep 1
timex rdblog -U -a /dev/rdsk/c21t49d0s1
sleep 1
timex rdblog -U -a /dev/rdsk/c21t49d0s3
sleep 1
timex rdblog -U -a /dev/rdsk/c21t49d0s4
sleep 1
timex rdblog -U -a /dev/rdsk/c21t49d0s5
sleep 1
timex rdblog -U -a /dev/rdsk/c21t49d0s6
sleep 1
timex rdblog -U -a /dev/rdsk/c21t50d0s1
sleep 1
timex rdblog -U -a /dev/rdsk/c21t50d0s3
sleep 1
timex rdblog -U -a /dev/rdsk/c21t50d0s4
sleep 1
timex rdblog -U -a /dev/rdsk/c21t50d0s5
sleep 1
timex rdblog -U -a /dev/rdsk/c21t50d0s6
sleep 1
timex rdblog -U -a /dev/rdsk/c21t51d0s1
sleep 1
timex rdblog -U -a /dev/rdsk/c21t51d0s3
sleep 1
timex rdblog -U -a /dev/rdsk/c21t51d0s4
sleep 1
timex rdblog -U -a /dev/rdsk/c21t51d0s5
sleep 1
timex rdblog -U -a /dev/rdsk/c21t51d0s6
sleep 1
timex rdblog -U -a /dev/rdsk/c21t52d0s1
sleep 1
timex rdblog -U -a /dev/rdsk/c21t52d0s3
sleep 1
timex rdblog -U -a /dev/rdsk/c21t52d0s4
sleep 1
timex rdblog -U -a /dev/rdsk/c21t52d0s5
sleep 1
timex rdblog -U -a /dev/rdsk/c21t52d0s6
sleep 1

******** mktmplog.sh ********
:
set -x
date
rm /rdbptc/tpcc80/SYS/rdblogmanage
timex rdblog -l

LOG_IX=/dev/vx/rdsk/vola03_02
LOG_BI=/dev/vx/rdsk/vola03_01
LOG_AI=/dev/vx/rdsk/vola01_01  #for 4Gover

timex rdblog -G -t -c 6600M -io 2048 $LOG_IX
$LOG_BI $LOG_AI 1000M 7000M 340
******** 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]
 *                              97-03-11
 *                                              -
make_alpha_string                       ,              .
 *          ( ORACLE     )
*/

#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: l_ID		(main	) */
		char		*EnvGetl_ID;
		int
		l_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,NEWORDER_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%02
d%02d%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_IWGRP|S_IRO
TH ))==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
*/
                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",
                 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_yt
d_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_IWGRP|S_IRO
TH ))==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_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",
                                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_state,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_next_o_id_3,d_n
ext_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_balance = -10.00;
                /* record       ,          */
                c_ytd_payment = 10.00;
                /* record       ,          */
                c_payment_cnt = 1;          /*
1997.04.24              */
                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_TRUNC,

S_IRUSR|S_IWUSR|S_IRGRP|S_IWGRP|S_IRO
TH ))==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)                 ;
            stmcpy(customer_cp->c_first   ,c_first
,len_i) ;
            stmcpy(customer_cp->c_middle
,c_middle  ,2    ) ;
            len_i = strlen(c_last)                  ;
            stmcpy(customer_cp->c_last    ,c_last
,len_i) ;

            len_i = strlen(c_street_1)              ;
            stmcpy(customer_cp-
>c_street_1,c_street_1,len_i) ;
            len_i = strlen(c_street_2)              ;
            stmcpy(customer_cp-
>c_street_2,c_street_2,len_i) ;
            len_i = strlen(c_city)                  ;
            stmcpy(customer_cp->c_city    ,c_city
,len_i) ;
            stmcpy(customer_cp->c_state   ,c_state
,2    ) ;
            stmcpy(customer_cp->c_zip     ,c_zip
,9    ) ;
            stmcpy(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_TRUNC,

S_IRUSR|S_IWUSR|S_IRGRP|S_IWGRP|S_IRO
TH ))==NULL){
                                printf("wttpcc:       : %s:
\n" , filename2);
                                exit(1);
                }

                /* 1997-02-18  fprintf -> sprintf + fwrite
*/
     /* HISTORY                          */

     history_ap = (char
*)malloc((size_t)(HISTORY_SIZE*HISTORY_CO
UNT)  );
     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_TRUNC,

S_IRUSR|S_IWUSR|S_IRGRP|S_IWGRP|S_IRO
TH ))==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_TRUNC,

S_IRUSR|S_IWUSR|S_IRGRP|S_IWGRP|S_IRO
TH ))==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",

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))
                                                /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;


        } 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);
    strcpy(&zip[4], "11111");       /* 1997.04.24 */


                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 <= mum) &&
(rnum <= 51)) {
                                str[i] = 'A' +
rnum - 26;

                    /* 52-61    ,
'0'==x30 (52:0, 53:1, .. , 61:9) */
                    } else if ((52 <= mum) &&
(rnum <= 61)) {
                                str[i] = '0' +
rnum - 52;
                    }
#else
                    /* 0-51                */
      mum = 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 <= mum) &&
(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' + mum;
#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 1680 tpmC 20170.63

Table	Rows	Data	Index	5% Space	8H Space	Total Space
Warehouse	1680	1718	0	85.9		1803.9
District	16800	16856	0	842.8		17698.8
Item	100000	14288	0	714.4		15002.4
New-order	15120000	1533672	454272	99397.2		2087341.2
History	50400000	3954048	0		759577.5	4713625.5
Orders	50400000	5054952	1283952		1217711.3	7556615.3
Customer	50400000	40321512	1229752	2077563.2		43628827.2
Order-line	503989400	54356864	0		10442020.9	64798884.9
Stock	168000000	67200280	0	3360014.0		70560294.0
DIRECTORY FILE		38912				38912.0
Dictionary		104139				104139.0
Totals		172597241	2967976	5538617.5	12419309.7	193523144.2

DB SPACE	199384578
Dynamic space	63365864.00
Static space	117737970.50
Free space	18280743.50
Daily growth	12172660.93
Daily spread	21752.11
180 day (KB)	2312732317.04
180 day (GB)	2205.59
Number of Archive LogFile	50
Size of Archive Log File (KB)	1766400.00
8 Hour Log (GB)	78.65
After image Log [GB] (mirror)	16.86
Before image Log [GB] (mirror)	3.90
Log index [GB] (mirror)	0.10

DISKS PRICED		
SIZE	Count	Capacity(GB)
2GB DISK	8	17.89
4GB DISK	7	27.40
9GB DISK	270	2276.50
Total	285	2321.79

Space Usage	
Usage	Size (GB)
180-day Space	2205.59
Root,swap,usr	13.94
Log	99.51
Total	2319.05

Appendix G: Distribution of Tables and Logs

Adapter Device	Using	Filename		Siz(MB)	Capacity	comment
SCSI c0t0d0	Operaiting System	/dev/rdsk/c0t0d0		--	2.2GB	
	SWAP			128.06		
	DIRECTORY FILE		s7	0.04		
SCSI c0t1d0	BI-LOG	/dev/rdsk/c0t1d0	--	1000.00	2.2GB	mirrored by VM (/dev/rdsk/vola03_01)
	IX-LOG		--	50.00		mirrored by VM (/dev/rdsk/vola03_02)
SCSI c4t5d0	BI-LOG(mirror)	/dev/rdsk/c4t5d0	--	1000.00	2.2GB	mirrored by VM (/dev/rdsk/vola03_01)
	IX-LOG(mirror)		--	50.00		mirrored by VM (/dev/rdsk/vola03_02)
SCSI c18t9d0	Striping N	/dev/rdsk/c18t9d0	s1	1195.84	2.2GB	It is the same with the c20t9d0s1
SCSI c18t10d0	Striping L	/dev/rdsk/c18t10d0	s1	1137.25	2.2GB	It is the same with the c20t10d0s1
SCSI c18t11d0	Striping L	/dev/rdsk/c18t11d0	s1	1137.25	2.2GB	It is the same with the c20t11d0s1
SCSI c18t12d0	Striping L	/dev/rdsk/c18t12d0	s1	1137.25	2.2GB	It is the same with the c20t12d0s1
SCSI c18t13d0	Striping L	/dev/rdsk/c18t13d0	s1	1137.25	2.2GB	It is the same with the c20t13d0s1
SCSI c4t2d0	Striping M	/dev/rdsk/c4t2d0	s1	1195.85	3.9GB	
	SWAP		s3	1959.26		
SCSI c4t3d0	Striping N	/dev/rdsk/c4t3d0	s1	1195.84	3.9GB	
	SWAP		s3	1959.26		
SCSI c4t4d0	Striping L	/dev/rdsk/c4t4d0	s1	1137.25	4.0GB	
SCSI c18t2d0	Striping L	/dev/rdsk/c18t2d0	s1	1137.25	4.0GB	It is the same with the c20t2d0s1
	SWAP		s3	2047.98		
SCSI c18t3d0	Striping L	/dev/rdsk/c18t3d0	s1	1137.25	4.0GB	It is the same with the c20t,Rd0s1
	SWAP		s3	2047.98		
SCSI c18t4d0	Striping L	/dev/rdsk/c18t4d0	s1	1137.25	3.9GB	It is the same with the c20t,Sd0s1
	SWAP		s3	1959.26		
SCSI c18t5d0	Striping M	/dev/rdsk/c18t5d0	s1	1195.85	4.0GB	It is the same with the c20t,Td0s1
	SWAP		s3	2047.98		
FCAL c2t64d0	Striping A	/dev/rdsk/c2t64d0	s1	1200.29	8.4GB	It is the same with the c6t64d0s1
	DICTIONNARY		s4	104.14		DICTIONARY
FCAL c2t65d0	Striping B	/dev/rdsk/c2t65d0	s1	1200.27	8.4GB	It is the same with the c6t65d0s1
FCAL c2t66d0	Striping C	/dev/rdsk/c2t66d0	s1	1141.67	8.4GB	It is the same with the c6t66d0s1
FCAL c2t67d0	ARC-LOG	/dev/rdsk/c2t67d0	s1	1725.00	8.4GB	It is the same with the c6t67d0s1
	ARC-LOG		s3	1725.00		It is the same with the c6t67d0s3
	ARC-LOG		s4	1725.00		It is the same with the c6t67d0s4
	ARC-LOG		s5	1725.00		It is the same with the c6t67d0s5
	ARC-LOG		s6	1725.00		It is the same with the c6t67d0s6

Adapter Device	Using	Filename		Siz(MB)	Capacity	comment
FCAL c2t68d0	Striping C	/dev/rdsk/c2t68d0	s1	1141.67	8.4GB	It is the same with the c6t68d0s1
FCAL c2t69d0	Striping C	/dev/rdsk/c2t69d0	s1	1141.67	8.4GB	It is the same with the c6t69d0s1
FCAL c2t70d0	Striping D	/dev/rdsk/c2t70d0	s1	1200.28	8.4GB	It is the same with the c6t70d0s1
FCAL c2t80d0	Striping B	/dev/rdsk/c2t80d0	s1	1200.27	8.4GB	It is the same with the c6t80d0s1
FCAL c2t81d0	Striping C	/dev/rdsk/c2t81d0	s1	1141.67	8.4GB	It is the same with the c6t81d0s1
FCAL c2t82d0	Striping C	/dev/rdsk/c2t82d0	s1	1141.67	8.4GB	It is the same with the c6t82d0s1
FCAL c2t83d0	Striping E	/dev/rdsk/c2t83d0	s1	1140.98	8.4GB	It is the same with the c6t83d0s1
FCAL c2t84d0	Striping F	/dev/rdsk/c2t84d0	s1	1140.28	8.4GB	It is the same with the c6t84d0s1
FCAL c2t85d0	Striping G	/dev/rdsk/c2t85d0	s1	1198.88	8.4GB	It is the same with the c6t85d0s1
FCAL c2t86d0	Striping H	/dev/rdsk/c2t86d0	s1	1198.88	8.4GB	It is the same with the c6t86d0s1
FCAL c2t96d0	Striping F	/dev/rdsk/c2t96d0	s1	1140.28	8.4GB	It is the same with the c6t96d0s1
FCAL c2t97d0	Striping F	/dev/rdsk/c2t97d0	s1	1140.28	8.4GB	It is the same with the c6t97d0s1
FCAL c2t98d0	Striping F	/dev/rdsk/c2t98d0	s1	1140.28	8.4GB	It is the same with the c6t98d0s1
FCAL c2t99d0	Striping F	/dev/rdsk/c2t99d0	s1	1140.28	8.4GB	It is the same with the c6t99d0s1
FCAL c2t100d0	Striping G	/dev/rdsk/c2t100d0	s1	1198.88	8.4GB	It is the same with the c6t100d0s1
FCAL c2t101d0	Striping H	/dev/rdsk/c2t101d0	s1	1198.88	8.4GB	It is the same with the c6t101d0s1
FCAL c2t102d0	Striping F	/dev/rdsk/c2t102d0	s1	1140.28	8.4GB	It is the same with the c6t102d0s1
FCAL c2t112d0	Striping F	/dev/rdsk/c2t112d0	s1	1140.28	8.4GB	It is the same with the c6t112d0s1
FCAL c2t113d0	Striping F	/dev/rdsk/c2t113d0	s1	1140.28	8.4GB	It is the same with the c6t113d0s1
FCAL c2t114d0	Striping F	/dev/rdsk/c2t114d0	s1	1140.28	8.4GB	It is the same with the c6t114d0s1
FCAL c2t115d0	Striping G	/dev/rdsk/c2t115d0	s1	1198.88	8.4GB	It is the same with the c6t115d0s1
FCAL c2t116d0	Striping H	/dev/rdsk/c2t116d0	s1	1198.88	8.4GB	It is the same with the c6t116d0s1
FCAL c2t117d0	Striping F	/dev/rdsk/c2t117d0	s1	1140.28	8.4GB	It is the same with the c6t117d0s1
FCAL c2t118d0	Striping F	/dev/rdsk/c2t118d0	s1	1140.28	8.4GB	It is the same with the c6t118d0s1
FCAL c10t0d0	Striping I	/dev/rdsk/c10t0d0	s1	1138.45	8.4GB	It is the same with the c11t0d0s1
FCAL c10t11d0	Striping I	/dev/rdsk/c10t1d0	s1	1138.45	8.4GB	It is the same with the c11t1d0s1
FCAL c10t2d0	Striping J	/dev/rdsk/c10t2d0	s1	1197.05	8.4GB	It is the same with the c11t2d0s1
FCAL c10t3d0	Striping K	/dev/rdsk/c10t3d0	s1	1197.04	8.4GB	It is the same with the c11t3d0s1
FCAL c10t4d0	Striping I	/dev/rdsk/c10t4d0	s1	1138.45	8.4GB	It is the same with the c11t4d0s1

Adapter Device	Using	Filename		Siz(MB)	Capacity	comment
FCAL c10t5d0	Striping I	/dev/rdsk/c10t5d0	s1	1138.45	8.4GB	It is the same with the c11t5d0s1
FCAL c10t6d0	Striping I	/dev/rdsk/c10t6d0	s1	1138.45	8.4GB	It is the same with the c11t6d0s1
FCAL c10t16d0	Striping I	/dev/rdsk/c10t16d0	s1	1138.45	8.4GB	It is the same with the c11t16d0s1
FCAL c10t17d0	Striping J	/dev/rdsk/c10t17d0	s1	1197.05	8.4GB	It is the same with the c11t17d0s1
FCAL c10t18d0	Striping K	/dev/rdsk/c10t18d0	s1	1197.04	8.4GB	It is the same with the c11t18d0s1
FCAL c10t19d0	Striping I	/dev/rdsk/c10t19d0	s1	1138.45	8.4GB	It is the same with the c11t19d0s1
FCAL c10t20d0	Striping I	/dev/rdsk/c10t20d0	s1	1138.45	8.4GB	It is the same with the c11t20d0s1
FCAL c10t21d0	Striping I	/dev/rdsk/c10t21d0	s1	1138.45	8.4GB	It is the same with the c11t21d0s1
FCAL c10t22d0	Striping I	/dev/rdsk/c10t22d0	s1	1138.45	8.4GB	It is the same with the c11t22d0s1
FCAL c10t32d0	Striping J	/dev/rdsk/c10t32d0	s1	1197.05	8.4GB	It is the same with the c11t32d0s1
FCAL c10t33d0	Striping K	/dev/rdsk/c10t33d0	s1	1197.04	8.4GB	It is the same with the c11t33d0s1
FCAL c10t34d0	Striping I	/dev/rdsk/c10t34d0	s1	1138.45	8.4GB	It is the same with the c11t34d0s1
FCAL c10t35d0	Striping I	/dev/rdsk/c10t35d0	s1	1138.45	8.4GB	It is the same with the c11t35d0s1
FCAL c10t36d0	Striping I	/dev/rdsk/c10t36d0	s1	1138.45	8.4GB	It is the same with the c11t36d0s1
FCAL c10t37d0	Striping I	/dev/rdsk/c10t37d0	s1	1138.45	8.4GB	It is the same with the c11t37d0s1
FCAL c10t38d0	Striping J	/dev/rdsk/c10t38d0	s1	1197.05	8.4GB	It is the same with the c11t38d0s1
FCAL c10t48d0	Striping K	/dev/rdsk/c10t48d0	s1	1197.04	8.4GB	It is the same with the c11t48d0s1
FCAL c10t49d0	Striping I	/dev/rdsk/c10t49d0	s1	1138.45	8.4GB	It is the same with the c11t49d0s1
FCAL c10t50d0	Striping I	/dev/rdsk/c10t50d0	s1	1138.45	8.4GB	It is the same with the c11t50d0s1
FCAL c10t51d0	Striping I	/dev/rdsk/c10t51d0	s1	1138.45	8.4GB	It is the same with the c11t51d0s1
FCAL c10t52d0	Striping I	/dev/rdsk/c10t52d0	s1	1138.45	8.4GB	It is the same with the c11t52d0s1
FCAL c10t53d0	Striping J	/dev/rdsk/c10t53d0	s1	1197.05	8.4GB	It is the same with the c11t53d0s1
FCAL c10t54d0	Striping K	/dev/rdsk/c10t54d0	s1	1197.04	8.4GB	It is the same with the c11t54d0s1
FCAL c16t64d0	Striping L	/dev/rdsk/c16t64d0	s1	1137.25	8.4GB	It is the same with the c17t64d0s1
FCAL c16t65d0	Striping L	/dev/rdsk/c16t65d0	s1	1137.25	8.4GB	It is the same with the c17t65d0s1
FCAL c16t66d0	Striping L	/dev/rdsk/c16t66d0	s1	1137.25	8.4GB	It is the same with the c17t66d0s1
FCAL c16t67d0	Striping L	/dev/rdsk/c16t67d0	s1	1137.25	8.4GB	It is the same with the c17t67d0s1
FCAL c16t68d0	Striping M	/dev/rdsk/c16t68d0	s1	1195.85	8.4GB	It is the same with the c17t68d0s1
FCAL c16t69d0	Striping N	/dev/rdsk/c16t69d0	s1	1195.84	8.4GB	It is the same with the c17t69d0s1

Adapter Device	Using	Filename		Siz(MB)	Capacity	comment
FCAL c16t70d0	Striping L	/dev/rdsk/c16t70d0	s1	1137.25	8.4GB	It is the same with the c17t70d0s1
FCAL c16t80d0	Striping L	/dev/rdsk/c16t80d0	s1	1137.25	8.4GB	It is the same with the c17t80d0s1
FCAL c16t81d0	Striping L	/dev/rdsk/c16t81d0	s1	1137.25	8.4GB	It is the same with the c17t81d0s1
FCAL c16t82d0	Striping L	/dev/rdsk/c16t82d0	s1	1137.25	8.4GB	It is the same with the c17t82d0s1
FCAL c16t83d0	Striping M	/dev/rdsk/c16t83d0	s1	1195.85	8.4GB	It is the same with the c17t83d0s1
FCAL c16t84d0	Striping N	/dev/rdsk/c16t84d0	s1	1195.84	8.4GB	It is the same with the c17t84d0s1
FCAL c16t85d0	Striping L	/dev/rdsk/c16t85d0	s1	1137.25	8.4GB	It is the same with the c17t85d0s1
FCAL c16t86d0	Striping L	/dev/rdsk/c16t86d0	s1	1137.25	8.4GB	It is the same with the c17t86d0s1
FCAL c16t96d0	Striping L	/dev/rdsk/c16t96d0	s1	1137.25	8.4GB	It is the same with the c17t96d0s1
FCAL c16t97d0	Striping L	/dev/rdsk/c16t97d0	s1	1137.25	8.4GB	It is the same with the c17t97d0s1
FCAL c16t98d0	Striping M	/dev/rdsk/c16t98d0	s1	1195.85	8.4GB	It is the same with the c17t98d0s1
FCAL c16t99d0	Striping N	/dev/rdsk/c16t99d0	s1	1195.84	8.4GB	It is the same with the c17t99d0s1
FCAL c16t100d0	Striping L	/dev/rdsk/c16t100d0	s1	1137.25	8.4GB	It is the same with the c17t100d0s1
FCAL c16t101d0	Striping L	/dev/rdsk/c16t101d0	s1	1137.25	8.4GB	It is the same with the c17t101d0s1
FCAL c16t102d0	Striping L	/dev/rdsk/c16t102d0	s1	1137.25	8.4GB	It is the same with the c17t102d0s1
FCAL c16t112d0	Striping L	/dev/rdsk/c16t112d0	s1	1137.25	8.4GB	It is the same with the c17t112d0s1
FCAL c16t113d0	Striping M	/dev/rdsk/c16t113d0	s1	1195.85	8.4GB	It is the same with the c17t113d0s1
FCAL c16t114d0	Striping N	/dev/rdsk/c16t114d0	s1	1195.84	8.4GB	It is the same with the c17t114d0s1
FCAL c16t115d0	Striping L	/dev/rdsk/c16t115d0	s1	1137.25	8.4GB	It is the same with the c17t115d0s1
FCAL c16t116d0	Striping L	/dev/rdsk/c16t116d0	s1	1137.25	8.4GB	It is the same with the c17t116d0s1
FCAL c16t117d0	Striping L	/dev/rdsk/c16t117d0	s1	1137.25	8.4GB	It is the same with the c17t117d0s1
FCAL c16t118d0	Striping L	/dev/rdsk/c16t118d0	s1	1137.25	8.4GB	It is the same with the c17t118d0s1
FCAL c12t0d0	Striping M	/dev/rdsk/c12t0d0	s1	1195.85	8.4GB	It is the same with the c13t0d0s1
FCAL c12t1d0	Striping N	/dev/rdsk/c12t1d0	s1	1195.84	8.4GB	It is the same with the c13t1d0s1
FCAL c12t2d0	Striping L	/dev/rdsk/c12t2d0	s1	1137.25	8.4GB	It is the same with the c13t2d0s1
FCAL c12t3d0	Striping L	/dev/rdsk/c12t3d0	s1	1137.25	8.4GB	It is the same with the c13t3d0s1
FCAL c12t4d0	Striping L	/dev/rdsk/c12t4d0	s1	1137.25	8.4GB	It is the same with the c13t4d0s1
FCAL c12t5d0	Striping L	/dev/rdsk/c12t5d0	s1	1137.25	8.4GB	It is the same with the c13t5d0s1
FCAL c12t6d0	Striping M	/dev/rdsk/c12t6d0	s1	1195.85	8.4GB	It is the same with the c13t6d0s1

Adapter Device	Using	Filename		Siz(MB)	Capacity	comment
FCAL c12t16d0	Striping N	/dev/rdsk/c12t16d0	s1	1195.84	8.4GB	It is the same with the c13t16d0s1
FCAL c12t17d0	Striping L	/dev/rdsk/c12t17d0	s1	1137.25	8.4GB	It is the same with the c13t17d0s1
FCAL c12t18d0	Striping L	/dev/rdsk/c12t18d0	s1	1137.25	8.4GB	It is the same with the c13t18d0s1
FCAL c12t19d0	Striping L	/dev/rdsk/c12t19d0	s1	1137.25	8.4GB	It is the same with the c13t19d0s1
FCAL c12t20d0	Striping L	/dev/rdsk/c12t20d0	s1	1137.25	8.4GB	It is the same with the c13t20d0s1
FCAL c12t21d0	Striping M	/dev/rdsk/c12t21d0	s1	1195.85	8.4GB	It is the same with the c13t21d0s1
FCAL c12t22d0	Striping N	/dev/rdsk/c12t22d0	s1	1195.84	8.4GB	It is the same with the c13t22d0s1
FCAL c12t32d0	Striping L	/dev/rdsk/c12t32d0	s1	1137.25	8.4GB	It is the same with the c13t32d0s1
FCAL c12t33d0	ARC-LOG	/dev/rdsk/c12t33d0	s1	1725.00	8.4GB	It is the same with the c13t33d0s1
	ARC-LOG		s3	1725.00		It is the same with the c13t33d0s3
	ARC-LOG		s4	1725.00		It is the same with the c13t33d0s4
	ARC-LOG		s5	1725.00		It is the same with the c13t33d0s5
	ARC-LOG		s6	1725.00		It is the same with the c13t33d0s6
FCAL c12t34d0	Striping L	/dev/rdsk/c12t34d0	s1	1137.25	8.4GB	It is the same with the c13t34d0s1
FCAL c12t35d0	Striping L	/dev/rdsk/c12t35d0	s1	1137.25	8.4GB	It is the same with the c13t35d0s1
FCAL c12t36d0	Striping M	/dev/rdsk/c12t36d0	s1	1195.85	8.4GB	It is the same with the c13t36d0s1
FCAL c12t37d0	Striping N	/dev/rdsk/c12t37d0	s1	1195.84	8.4GB	It is the same with the c13t37d0s1
FCAL c12t38d0	Striping L	/dev/rdsk/c12t38d0	s1	1137.25	8.4GB	It is the same with the c13t38d0s1
FCAL c12t48d0	Striping L	/dev/rdsk/c12t48d0	s1	1137.25	8.4GB	It is the same with the c13t48d0s1
FCAL c12t49d0	Striping L	/dev/rdsk/c12t49d0	s1	1137.25	8.4GB	It is the same with the c13t49d0s1
FCAL c12t50d0	Striping L	/dev/rdsk/c12t50d0	s1	1137.25	8.4GB	It is the same with the c13t50d0s1
FCAL c12t51d0	Striping M	/dev/rdsk/c12t51d0	s1	1195.85	8.4GB	It is the same with the c13t51d0s1
FCAL c12t52d0	Striping N	/dev/rdsk/c12t52d0	s1	1195.84	8.4GB	It is the same with the c13t52d0s1
FCAL c12t53d0	Striping L	/dev/rdsk/c12t53d0	s1	1137.25	8.4GB	It is the same with the c13t53d0s1
FCAL c12t54d0	Striping L	/dev/rdsk/c12t54d0	s1	1137.25	8.4GB	It is the same with the c13t54d0s1
FCAL c14t64d0	Striping L	/dev/rdsk/c14t64d0	s1	1137.25	8.4GB	It is the same with the c15t64d0s1
FCAL c14t65d0	Striping L	/dev/rdsk/c14t65d0	s1	1137.25	8.4GB	It is the same with the c15t65d0s1
FCAL c14t66d0	Striping M	/dev/rdsk/c14t66d0	s1	1195.85	8.4GB	It is the same with the c15t66d0s1
FCAL c14t67d0	Striping N	/dev/rdsk/c14t67d0	s1	1195.84	8.4GB	It is the same with the c15t67d0s1

Adapter Device	Using	Filename		Siz(MB)	Capacity	comment
FCAL c14t68d0	Striping L	/dev/rdsk/c14t68d0	s1	1137.25	8.4GB	It is the same with the c15t68d0s1
FCAL c14t69d0	Striping L	/dev/rdsk/c14t69d0	s1	1137.25	8.4GB	It is the same with the c15t69d0s1
FCAL c14t70d0	Striping L	/dev/rdsk/c14t70d0	s1	1137.25	8.4GB	It is the same with the c15t70d0s1
FCAL c14t80d0	Striping L	/dev/rdsk/c14t80d0	s1	1137.25	8.4GB	It is the same with the c15t80d0s1
FCAL c14t81d0	Striping M	/dev/rdsk/c14t81d0	s1	1195.85	8.4GB	It is the same with the c15t81d0s1
FCAL c14t82d0	Striping N	/dev/rdsk/c14t82d0	s1	1195.84	8.4GB	It is the same with the c15t82d0s1
FCAL c14t83d0	Striping L	/dev/rdsk/c14t83d0	s1	1137.25	8.4GB	It is the same with the c15t83d0s1
FCAL c14t84d0	Striping L	/dev/rdsk/c14t84d0	s1	1137.25	8.4GB	It is the same with the c15t84d0s1
FCAL c14t85d0	Striping L	/dev/rdsk/c14t85d0	s1	1137.25	8.4GB	It is the same with the c15t85d0s1
FCAL c14t86d0	Striping L	/dev/rdsk/c14t86d0	s1	1137.25	8.4GB	It is the same with the c15t86d0s1
FCAL c14t96d0	Striping M	/dev/rdsk/c14t96d0	s1	1195.85	8.4GB	It is the same with the c15t96d0s1
FCAL c14t97d0	Striping N	/dev/rdsk/c14t97d0	s1	1195.84	8.4GB	It is the same with the c15t97d0s1
FCAL c14t98d0	Striping L	/dev/rdsk/c14t98d0	s1	1137.25	8.4GB	It is the same with the c15t98d0s1
FCAL c14t99d0	Striping L	/dev/rdsk/c14t99d0	s1	1137.25	8.4GB	It is the same with the c15t99d0s1
FCAL c14t100d0	Striping L	/dev/rdsk/c14t100d0	s1	1137.25	8.4GB	It is the same with the c15t100d0s1
FCAL c14t101d0	Striping L	/dev/rdsk/c14t101d0	s1	1137.25	8.4GB	It is the same with the c15t101d0s1
FCAL c14t102d0	Striping M	/dev/rdsk/c14t102d0	s1	1195.85	8.4GB	It is the same with the c15t102d0s1
FCAL c14t112d0	Striping N	/dev/rdsk/c14t112d0	s1	1195.84	8.4GB	It is the same with the c15t112d0s1
FCAL c14t113d0	Striping L	/dev/rdsk/c14t113d0	s1	1137.25	8.4GB	It is the same with the c15t113d0s1
FCAL c14t114d0	Striping L	/dev/rdsk/c14t114d0	s1	1137.25	8.4GB	It is the same with the c15t114d0s1
FCAL c14t115d0	Striping L	/dev/rdsk/c14t115d0	s1	1137.25	8.4GB	It is the same with the c15t115d0s1
FCAL c14t116d0	Striping L	/dev/rdsk/c14t116d0	s1	1137.25	8.4GB	It is the same with the c15t116d0s1
FCAL c14t117d0	Striping M	/dev/rdsk/c14t117d0	s1	1195.85	8.4GB	It is the same with the c15t117d0s1
FCAL c14t118d0	Striping N	/dev/rdsk/c14t118d0	s1	1195.84	8.4GB	It is the same with the c15t118d0s1
FCAL c21t0d0	Striping L	/dev/rdsk/c21t0d0	s1	1137.25	8.4GB	It is the same with the c22t0d0s1
FCAL c21t1d0	Striping L	/dev/rdsk/c21t1d0	s1	1137.25	8.4GB	It is the same with the c22t1d0s1
FCAL c21t2d0	Striping L	/dev/rdsk/c21t2d0	s1	1137.25	8.4GB	It is the same with the c22t2d0s1
FCAL c21t3d0	Striping L	/dev/rdsk/c21t3d0	s1	1137.25	8.4GB	It is the same with the c22t3d0s1
FCAL c21t4d0	Striping M	/dev/rdsk/c21t4d0	s1	1195.85	8.4GB	It is the same with the c22t4d0s1

Adapter Device	Using	Filename		Siz(MB)	Capacity	comment
FCAL c21t5d0	Striping N	/dev/rdsk/c21t5d0	s1	1195.84	8.4GB	It is the same with the c22t5d0s1
FCAL c21t6d0	Striping L	/dev/rdsk/c21t6d0	s1	1137.25	8.4GB	It is the same with the c22t6d0s1
FCAL c21t16d0	Striping L	/dev/rdsk/c21t16d0	s1	1137.25	8.4GB	It is the same with the c22t16d0s1
FCAL c21t17d0	Striping L	/dev/rdsk/c21t17d0	s1	1137.25	8.4GB	It is the same with the c22t17d0s1
FCAL c21t18d0	Striping L	/dev/rdsk/c21t18d0	s1	1137.25	8.4GB	It is the same with the c22t18d0s1
FCAL c21t19d0	Striping M	/dev/rdsk/c21t19d0	s1	1195.85	8.4GB	It is the same with the c22t19d0s1
FCAL c21t20d0	Striping N	/dev/rdsk/c21t20d0	s1	1195.84	8.4GB	It is the same with the c22t20d0s1
FCAL c21t21d0	Striping L	/dev/rdsk/c21t21d0	s1	1137.25	8.4GB	It is the same with the c22t21d0s1
FCAL c21t22d0	Striping L	/dev/rdsk/c21t22d0	s1	1137.25	8.4GB	It is the same with the c22t22d0s1
FCAL c21t32d0	Striping L	/dev/rdsk/c21t32d0	s1	1137.25	8.4GB	It is the same with the c22t32d0s1
FCAL c21t33d0	Striping L	/dev/rdsk/c21t33d0	s1	1137.25	8.4GB	It is the same with the c22t33d0s1
FCAL c21t34d0	ARC-LOG	/dev/rdsk/c21t34d0	s1	1725.00	8.4GB	It is the same with the c22t34d0s1
	ARC-LOG		s3	1725.00		It is the same with the c22t34d0s3
	ARC-LOG		s4	1725.00		It is the same with the c22t34d0s4
	ARC-LOG		s5	1725.00		It is the same with the c22t34d0s5
	ARC-LOG		s6	1725.00		It is the same with the c22t34d0s6
FCAL c21t35d0	ARC-LOG	/dev/rdsk/c21t35d0	s1	1725.00	8.4GB	It is the same with the c22t35d0s1
	ARC-LOG		s3	1725.00		It is the same with the c22t35d0s3
	ARC-LOG		s4	1725.00		It is the same with the c22t35d0s4
	ARC-LOG		s5	1725.00		It is the same with the c22t35d0s5
	ARC-LOG		s6	1725.00		It is the same with the c22t35d0s6
FCAL c21t36d0	ARC-LOG	/dev/rdsk/c21t36d0	s1	1725.00	8.4GB	It is the same with the c22t36d0s1
	ARC-LOG		s3	1725.00		It is the same with the c22t36d0s3
	ARC-LOG		s4	1725.00		It is the same with the c22t36d0s4
	ARC-LOG		s5	1725.00		It is the same with the c22t36d0s5
	ARC-LOG		s6	1725.00		It is the same with the c22t36d0s6
FCAL c21t37d0	ARC-LOG	/dev/rdsk/c21t37d0	s1	1725.00	8.4GB	It is the same with the c22t37d0s1
	ARC-LOG		s3	1725.00		It is the same with the c22t37d0s3
	ARC-LOG		s4	1725.00		It is the same with the c22t37d0s4
	ARC-LOG		s5	1725.00		It is the same with the c22t37d0s5

Adapter Device	Using	Filename		Siz(MB)	Capacity	comment
	ARC-LOG		s6	1725.00		It is the same with the c22t37d0s6
FCAL c21t38d0	Striping L	/dev/rdsk/c21t38d0	s1	1137.25	8.4GB	It is the same with the c22t38d0s1
FCAL c21t48d0	Striping C	/dev/rdsk/c21t48d0	s1	1141.67	8.4GB	It is the same with the c22t48d0s1
FCAL c21t49d0	ARC-LOG	/dev/rdsk/c21t49d0	s1	1725.00	8.4GB	It is the same with the c22t49d0s1
	ARC-LOG		s3	1725.00		It is the same with the c22t49d0s3
	ARC-LOG		s4	1725.00		It is the same with the c22t49d0s4
	ARC-LOG		s5	1725.00		It is the same with the c22t49d0s5
	ARC-LOG		s6	1725.00		It is the same with the c22t49d0s6
FCAL c21t50d0	ARC-LOG	/dev/rdsk/c21t50d0	s1	1725.00	8.4GB	It is the same with the c22t50d0s1
	ARC-LOG		s3	1725.00		It is the same with the c22t50d0s3
	ARC-LOG		s4	1725.00		It is the same with the c22t50d0s4
	ARC-LOG		s5	1725.00		It is the same with the c22t50d0s5
	ARC-LOG		s6	1725.00		It is the same with the c22t50d0s6
FCAL c21t51d0	ARC-LOG	/dev/rdsk/c21t51d0	s1	1725.00	8.4GB	It is the same with the c22t51d0s1
	ARC-LOG		s3	1725.00		It is the same with the c22t51d0s3
	ARC-LOG		s4	1725.00		It is the same with the c22t51d0s4
	ARC-LOG		s5	1725.00		It is the same with the c22t51d0s5
	ARC-LOG		s6	1725.00		It is the same with the c22t51d0s6
FCAL c21t52d0	ARC-LOG	/dev/rdsk/c21t52d0	s1	1725.00	8.4GB	It is the same with the c22t52d0s1
	ARC-LOG		s3	1725.00		It is the same with the c22t52d0s3
	ARC-LOG		s4	1725.00		It is the same with the c22t52d0s4
	ARC-LOG		s5	1725.00		It is the same with the c22t52d0s5
	ARC-LOG		s6	1725.00		It is the same with the c22t52d0s6
FCAL c21t53d0	AI-LOG	/dev/rdsk/c21t53d0	--	7000.00	8.4GB	It is the same with the c22t53d0s1 mirrored by VM (/dev/rdsk/vola01_01)
FCAL c21t54d0	AI-LOG (mirror)	/dev/rdsk/c21t54d0	--	7000.00	8.4GB	It is the same with the c22t54d0s1 mirrored by VM (/dev/rdsk/vola01_01)

Striping A	
Table name	Size(Kbytes)
WareHouse	3108
Stock	460010
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792
Item	1435

Striping B	
Table name	Size(Kbytes)
WareHouse	3108
Stock	460002
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792
Item	1428

Striping C	
Table name	Size(Kbytes)
WareHouse	3108
Stock	400000
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792
Item	1428

Striping D	
Table name	Size(Kbytes)
WareHouse	3108
Stock	460010
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792
Item	1428

Striping E	
Table name	Size(Kbytes)
WareHouse	3108
Stock	400000
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792
Item	716

Striping F	
Table name	Size(Kbytes)
WareHouse	3108
Stock	400000
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792

Striping G	
Table name	Size(Kbytes)
WareHouse	3108
Stock	460010
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792

Striping H	
Table name	Size(Kbytes)
WareHouse	3108
Stock	460002
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792

Striping I	
Table name	Size(Kbytes)
District	1231
Stock	400000
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792

Striping J	
Table name	Size(Kbytes)
District	1231
Stock	460010
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792

Striping K	
Table name	Size(Kbytes)
District	1231
Stock	460002
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792

Striping L	
Table name	Size(Kbytes)
Stock	400000
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792

Striping M	
Table name	Size(Kbytes)
Stock	460010
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792

Striping N	
Table name	Size(Kbytes)
Stock	460002
NewOrder	13504
NewOrder Index	6144
Orders	38461
Orders Index	15424
History	29492
Customer	252010
Customer Index	7712
OrderLine	401792

Appendix H: Price Quotes

98-07-10 17:11   FROM FUJITSU SOFTWARE   P02

HAL Computer Systems   ID:1-408-341-6401   JUL 09'98   15:19 No.002 P.02

Sun microsystems

Sales Quotation

Quote Number: GTA-ER-31207-A

Quote Date: 07/09/98 (Quote Valid For 60 Days)

To: Deanna Hurd
HaL Computer Systems
1315 Dell Avenue
Campbell, CA 95008
Ph/Fax: 408-341-6518 / 408-341-6901

From: Eric Romalfanger
General Territory
1842 North Shoreline Blvd.
Mountain View, CA 94043
Ph/Fax: (650) 960-4262/(650) 967-6586

We are pleased to quote as follows

Approved By: ____________

Reference	Credit Terms	F.O.B.	Page
	Net 30 Days	Point-of-origin	1 of 2

Item	Product Number	Description	Qty	Unit List Price	Disc.	Unit Net Price	Extended Net Price (US $)
		A5000 array:					
1.0	SG-XARY012A-254G	254.8GB Sun StorEdge A5000 (2 arrays with 14 x 9.1GB, 7200rpm FC-AL drives) 2 FC-100 hubs (3 GBICs each) w/ 2 15 meter Fibre Channel cables, mounted in Enterprise expansion cabinet	1	$116,900.00	40.000	$70,140.00	$70,140.00
2.0	SG-XARY011A-127G	127.4-Gbyte Sun StorEdge A5000 (14 x 9.1GB, 7200rpm FC-AL drives) rackmount configuration mounted on bracket w/ 2 meter Fibre Channel cable	1	$63,000.00	40.000	$37,800.00	$37,800.00
3.0	X6731A	FC AL 100MB/sec GBIC module	1	$600.00	40.000	$360.00	$360.00
4.0	X6730A	FC-AL 100MB/sec Dual Channel SBus host adapter w/1 GBIC module	1	$2,700.00	40.000	$1,620.00	$1,620.00
5.0	X6732A	FC AL 100MB/s 7 slot hub w/ no GBIC modules	1	$1,000.00	40.000	$600.00	$600.00
6.0	X978A	15-meter Fibre Optic Cable *The A5000 array is supported on the Enterprise 450, 3000, 4000, 5000, 6000 class of servers.*	1	$175.00	N/A	$175.00	$175.00

[illegible]

Accepted By:______________________ Title:____________________ Date:__________

Note: Not valid unless accompanied by a hardcopy Purchase Order.

QUOTEMASTER

FROM 1-408-341-6401   98⁰07月10日 08時21分   P02

98-07-10 17:11 FROM FUJITSU SOFTWARE P03

HAL Computer Systems ID:1-408-341-6401 JUL 09 98 15:19 No.002 P.03

Sun microsystems

Sales Quotation

Quote Number: GTA-ER-31207-A

Quote Date: 07/09/98 (Quote Valid For 63 Days)

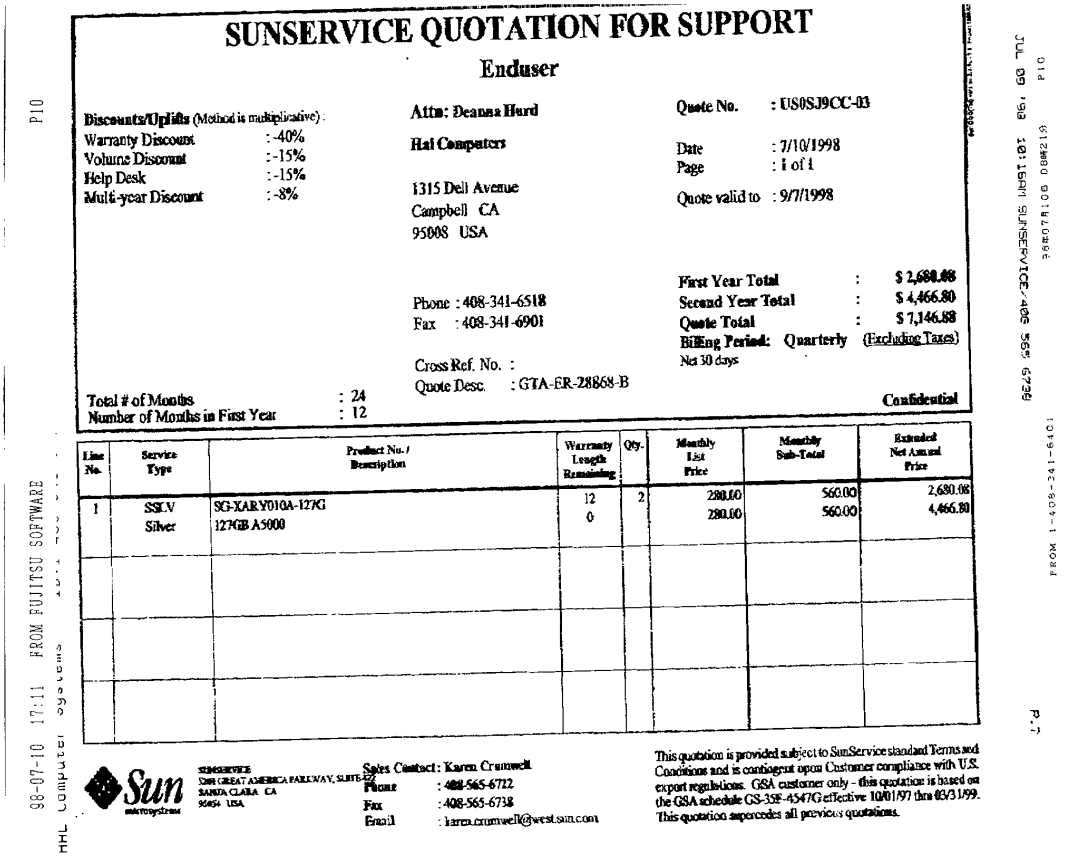
Page 2 of 2

Item	Product Number	Description	Qty	Unit List Price	Disc	Unit Net Price	Extended Net Price (US $)
		Quote Total:					$110,695.00

Accepted By:______________________ Title:__________________ Date:__________

Note: Not valid unless accompanied by a hardcopy Purchase Order.

FROM 1-408-341-6401 98年07月10日 08時21分 P03



SUNSERVICE QUOTATION FOR SUPPORT

Enduser

Discounts/Uplifts (Method is multiplicative):
Warranty Discount : -40%
Volume Discount : -15%
Help Desk : -15%
Multi-year Discount : -8%

Attn: Deanna Hurd
Hal Computers
1315 Dell Avenue
Campbell CA
95008 USA

Phone : 408-341-6518
Fax : 408-341-6901

Cross Ref. No. :
Quote Desc. : GTA-ER-28868-B

Quote No. : US0SJ9CC-03
Date : 7/10/1998
Page : 1 of 1
Quote valid to : 9/7/1998

First Year Total : $ 2,680.00
Second Year Total : $ 4,466.80
Quote Total : $ 7,146.88
Billing Period: Quarterly (Excluding Taxes)
Net 30 days

Total # of Months : 24
Number of Months in First Year : 12

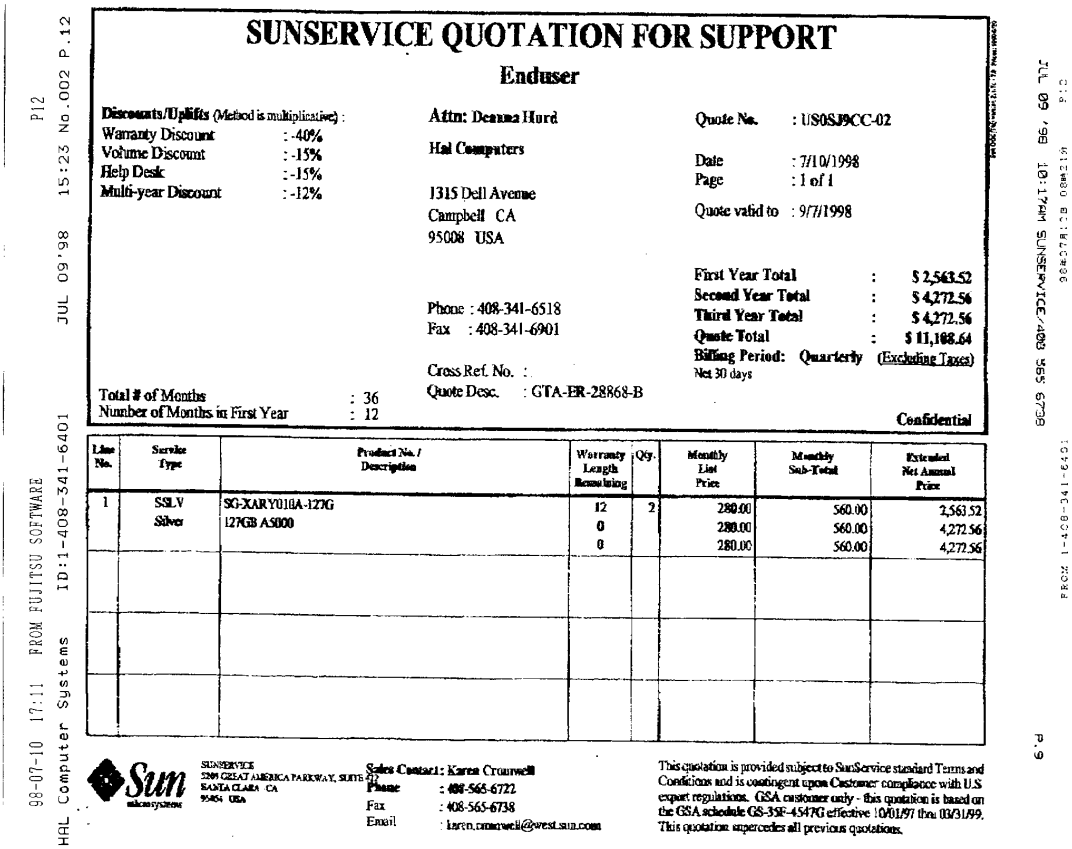
Confidential

Line No.	Service Type	Product No. / Description	Warranty Length Remaining	Qty.	Monthly List Price	Monthly Sub-Total	Extended Net Annual Price
1	SSLV Silver	SG-XARY010A-127G 127GB A5000	12 0	2	280.00 280.00	560.00 560.00	2,680.00 4,466.80

SUNSERVICE
3300 GREAT AMERICA PARKWAY, SUITE 500
SANTA CLARA CA
95054 USA

Sales Contact : Karen Crumwell
Phone : 408-565-6722
Fax : 408-565-6738
Email : karen.crumwell@west.sun.com

This quotation is provided subject to SunService standard Terms and Conditions and is contingent upon Customer compliance with U.S. export regulations. GSA customer only - this quotation is based on the GSA schedule GS-35F-4547G effective 10/01/97 thru 03/31/99. This quotation supercedes all previous quotations.



98-07-10 17:11 FROM FUJITSU SOFTWARE HAL Computer Systems ID:1-408-341-6401 JUL 09'98 15:23 No.002 P.12 P12

SUNSERVICE QUOTATION FOR SUPPORT

Enduser

Discounts/Uplifts (Method is multiplicative) :

Warranty Discount	: -40%
Volume Discount	: -15%
Help Desk	: -15%
Multi-year Discount	: -12%

Attn: Deanna Hurd

Hal Computers

1315 Dell Avenue
Campbell CA
95008 USA

Phone : 408-341-6518
Fax : 408-341-6901

Cross Ref. No. :
Quote Desc. : GTA-ER-28868-B

Quote No. : US0SJ9CC-02

Date : 7/10/1998
Page : 1 of 1

Quote valid to : 9/7/1998

First Year Total	:	**$ 2,563.52**
Second Year Total	:	**$ 4,272.56**
Third Year Total	:	**$ 4,272.56**
Quote Total	:	**$ 11,108.64**

Billing Period: Quarterly (Excluding Taxes)
Net 30 days

Total # of Months : 36
Number of Months in First Year : 12

Confidential

Line No.	Service Type	Product No. / Description	Warranty Length Remaining	Qty.	Monthly List Price	Monthly Sub-Total	Extended Net Annual Price
1	SSLV	SG-XARY01BA-127G	12	2	280.00	560.00	2,563.52
	Silver	127GB A5000	0		280.00	560.00	4,272.56
			0		280.00	560.00	4,272.56

Sun microsystems

SUNSERVICE
2200 GREAT AMERICA PARKWAY, SUITE 412
SANTA CLARA CA
95054 USA

Sales Contact: Karen Cromwell
Phone : 408-565-6722
Fax : 408-565-6738
Email : karen.cromwell@west.sun.com

This quotation is provided subject to SunService standard Terms and Conditions and is contingent upon Customer compliance with U.S. export regulations. GSA customer only - this quotation is based on the GSA schedule GS-35F-4547G effective 10/01/97 thru 09/31/99. This quotation supercedes all previous quotations.

JUL 09 '98 10:17AM SUNSERVICE/408 565 6738 P.9 FROM 1-408-341-6401

AL Computer Systems ID:1-408-341-6401 JUN 19'98 19:17 No.004 P.04

NETLUX

14180 Live Oak Ave., Unit E
Baldwin Park, Ca. 91760

1-800-789-1780
Phone#818-851-9737
Fax #818-851-9837

June 19, 1998

Deanna Hurd

Quotation

Quantity	*Part No.*	*Description*	*Unit Price*	*Total*
1	NX-H9+	8-port 10Mbps Ethernet Hub	$42.00	$ 42.00

Terms and Conditions:
FOB Origin
5 Year Warranty
Quote Valid for 90 Days

Sincerely,
Martin Parry
NETLUX

FROM 1 408 341 6401 98年06月19日 1:18pm P01

98-07-10 17:11 FROM FUJITSU SOFTWARE P16

HAL Computer Systems ID:1-408-341-6401 JUL 09'98 15:25 No.002 P.16

1304 South 51st Street, Richmond CA 94804
510-412-4330 x230
510-412-4343
http://www.ssp.com

Quote

Quote# 071099sm1
Prepared by: Sandesh S. Mutha

Bill To
HaL Computer Systems
Attn: Accounts Payable
1315 Dell Avenue
Campbell, CA 95008

Ship To
HaL Computer Systems
Attn:Deanna Hurd
1315 Dell Avenue
Campbell, CA 95008

Terms: Quote valid for 60 days - acceptance based on credit approval by SSP
Prices quoted do not include applicable freight or tax charges

Description	Part #	Quantity	Unit Expense	Ext Expense
8 port FE switch	NH2012	1	$2,100.00	$2,100.00
6 year advance exchange warranty	NH2012-6IA	1	$1,098.00	$1,098.00

Grand Total $3,198.00

If you have any questions or if you need additional information, please contact me. Thank you for your time and consideration.

Regards,

Sandesh S. Mutha

TOTAL P.01

FROM 1-408-341-6401 98年07月10日 08時21分 P16

HL Computer Systems ID:1-408-341-6401 JUN 19'98 19:17 No.004 P.05

1304 South 51st Street, Richmond CA 94804
510-412-4330 x230
510-412-4845
http://www.ssp.com

Quote

Date: 19 June 1998
Quote #: 061998sm3
Prepared by: Sandesh S. Mutha

Bill To
HaL Computer Systems
Attn: Accounts Payable
1315 Dell Avenue
Campbell, CA 95008

Ship To
HaL Computer Systems
Attn:Deanna Hurd
1315 Dell Avenue
Campbell, CA 95008

Terms: Quote valid for 90 days - acceptance based on credit approval by SSP
Prices quoted do not include applicable freight or tax charges

Description	Part #	Quantity	Unit Expense	Ext Expense
Wyse WY-55 terminal (green)	WY-55	1	$295.00	$295.00
Wyse WY-65 terminal (green)	WY-65	1	$407.00	$407.00
Keyboard		1	$100.00	$100.00
3 year warranty extension for WY-55 or WY-65 (5 year total)		1	$100.00	$100.00

Grand Total **$902.00**

If you have any questions or if you need additional information, please contact me. Thank you for your time and consideration.

Regards,

Sandesh S. Mutha

TOTAL P.01

FROM 1-408-341-6401 [illegible] P05

BEA Systems, Inc.
385 Moffett Park Drive
Sunnyvale, California 94089

Telephone: (408) 743-4000
Fax: (408) 734-9234

BEA

Enterprise Middleware Solutions

June 25, 1998

FUJITSU LIMITED
Tokyo, Japan

Dear Sirs:

Per your request I am enclosing the pricing quotation regarding TUXEDO 4.2.2 that you requested.

Tuxedo 4.2.2 is priced per server for an unlimited number of users. The pricing varies by server performance level and BEA defines seven server levels or classes for machines on which Tuxedo is installed. The tables below provide the pricing and class levels for Intel-based servers and Sun Microsystems servers.

This pricing quotation will be valid through September 30, 1998.

TUXEDO 4.2.x Packaging/Pricing

Machine Class	List price	Maintenance (5 x 8) per year	Maintenance (7 x 24) per year
Class 1	$1,440.00	$216.00	$316.80
Class 2	$4,200.00	$630.00	$924.00
Class 3	$9,000.00	$1,350.00	$1,980.00
Class 4	$18,000.00	$2,700.00	$3,960.00
Class 5	$26,400.00	$3,960.00	$5,808.00
Class 6	$58,800.00	$8,820.00	$12,936.00
Class 7	$94,800.00	$14,220.00	$20,856.00

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

BEA

ENTERPRISE MIDDLEWARE SOLUTIONS

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

PERFORMANCE METRICS INC.
TPC Certified Auditors

July 22, 1998

Mr. Kazuhiko Saito
Manager,
Development DEPT. I
MIDDLEWARE DIV.
SOFTWARE GROUP
Fujitsu Limited
140 Miyamoto
Numazu-shi Shizuoka, 410-0396 Japan

I have verified the TPC Benchmark™ C client/server for the following configuration:

Platform:	GP 7000 Model 600
Database Manager:	SymfoWARE Server for VLM 2.0
Operating System:	Solaris2.6
Transaction Manager:	Tuxedo 4.2

Server: GP 7000 Model 600				
CPU's	Memory	Disks	90% Response	tpmC
8 UltraSPARC @ 296 MHz	Main: 8 GB Cache: 2MB each	168 @ 9GB 7 @ 4GB 8 @ 2GB	1.63 Sec	20,170.63
9 Clients: GP 7000 Model 200				
1 UltraSPARC @ 248 MHz	Main: 1 GB Cache: 1MB	2 @ 2 GB	na	na

In my opinion, these performance results were produced in compliance with the TPC requirements for the benchmark. The following attributes of the benchmark were given special attention:

- The transactions were correctly implemented.

2229 Benita Dr. Suite 101, Rancho Cordova, CA 95670
(916) 635-2822 fax: (916) 858-0109 email: Lorna@PerfMetrics.com

PERFORMANCE METRICS INC.
TPC Certified Auditors

- The database files were properly sized and populated.
- The database was properly scaled with 1,680 warehouses.
- The ACID properties were met.
- The ACID tests were performed on the measured database.
- Input data was generated according to the specified percentages.
- Eight hours of mirrored log space was present on the tested system.
- Eight hours of growth space for the dynamic tables was present on the tested system.
- The data for the 180 day space calculation was verified; 102 extra 9GB disks were added to the priced configuration.
- The steady state portion of the test was 30 minutes.
- One checkpoint was taken before the measured interval.
- One checkpoint was taken during the measured interval.
- The checkpoints were verified to be clear of the guard zone.
- The system pricing was checked for major components and maintenance.

Auditor Notes:

All tables except HISTORY used hash addressing. The system was run for 8 hours to demonstrate no performance degradation was caused by the addition of rows to growing tables. In my opinion, the hashing architecture satisfies Clause 0.2.

Sincerely,

Tom Sawyer

Tom Sawyer
Auditor

2229 Benita Dr. Suite 101, Rancho Cordova, CA 95670
(916) 635-2822 fax: (916) 858-0109 email: Loma@PerfMetrics.com