TPC Benchmark C
Full Disclosure Report
IBM AS/400
AS/400e server model s40 with feature code 2208

September 1, 1998

Special Notices

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

- Application System/400
- AS/400
- AS/400e
- CICS
- DB2 for AS/400
- INT LNG ENV COBOL
- INT LNG ENV C
- IBM
- OS/400
- Structured Query Language/400 (SQL/400)

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

TPC Benchmark Trademark of the Transaction Processing Performance Council

Revised September 1, 1998

The information contained in this document is distributed on an AS IS basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item has been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environment do so at their own risk.

In this document, any references made to an IBM-licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such products, programming, or services in your country.

All performance data contained in this publication was obtained in a controlled environment, and therefore the results which may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data in their specific environment.

Abstract

This report documents the full disclosure information required by the TPC Benchmark™ C Standard Specification dated April 25, 1998 for measurements on the IBM AS/400e server model s40 with feature code 2208.

The software used on the AS/400 systems includes OS/400 Version 4, Release 3, Modification 0, DB2 for AS/400 Version 4, Release 3, Modification 0, INT LNG ENV COBOL OS/400 V4 R3, INT LNG ENV C OS/400 V4 R3, DB2 Query Manager and SQL Development Kit, CICS for OS/400 V4 R3, and Application Development Tools.

IBM AS/400e server model s40 with feature code 2208

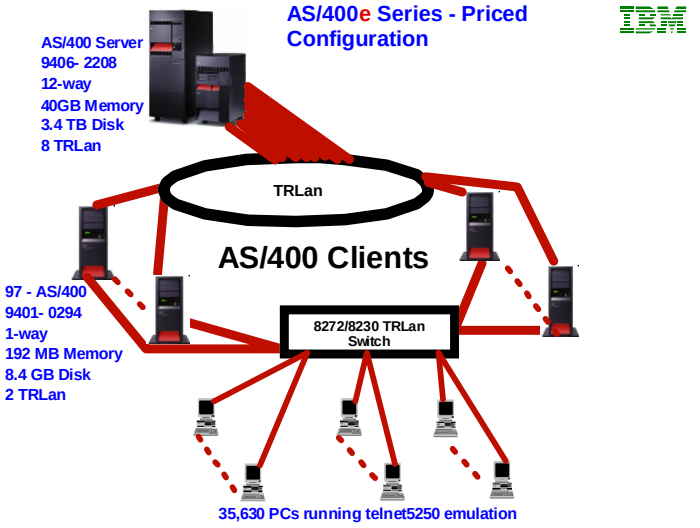
Company Name	System Name	Data Base Software	Operating System Software
IBM®	AS/400e server s40-2208 C/S	DB2 for AS/400 Version 4 Release 3	OS/400 Version 4 Release 3
Availability Date: September 11, 1998			

Total System Cost	TPC-C Throughput	Price/Performance
- Hardware - Software -5 Years Maintenance	Sustained maximum throughput of system running TPC-C expressed in transactions per minute	Total system cost/tpmC ($5,564,844/43,169.85)
$5,564,844	43,169.85 tpmC	$128.91 per tpmC

IBM	AS/400e server model s40 with feature code 2208	TPC-C Rev. 3.3 Report Date: September 1,, 1998	
Total System Cost	**TPC-C Throughput**	**Price/Performance**	**Availability Date**
$5,564,844	**43,169.85**	**$128.91**	**September 11, 1998**

Processors	Database Manager	Operating System	Other Software	Number of Users
1 AS/400e s40-2208 (12-way) 97 AS/400e 150-0294	DB2 for AS/400 Version 4 Release 3	OS/400 Version 4 Release 3	CICS for OS/400 V4 R3 INT LNG ENV COBOL OS/400 V4 R2 INT LNG ENV C OS/400 V4 R3	35,630

AS/400e server s40-2208 C/S Priced Configuration



System components:	Qty:	Description:
Server	1	
Processors	12	AS/400e server s40-2261
Memory		17 GB main memory
Disk controllers	21	6532 DASD IOP
	1	Integrated multifunction IOP
Disk drives	356	316 8589 MB DASD 40 17542 MB DASD
Total storage		3.4 TB
Clients (each):	97	
Processor	1	AS/400e server 150-0294
Memory		192 MB main memory
Disk controller	1	Integrated multifunction IOP
Disk	2	4194 MB DASD

Token-Ring Adapter (9249)	Included in S40 base price		1	$0		$0
2048 Optional MB Base Memory	8193	$39,424	4	$157,696		$157,696
Optical Link	2688	$2,000	6	$12,000		$12,000
Optical Bus Adapter	2695	$1,000	1	$1,000		$1,000
2048 MB Main Storage	3193	$45,056	16	$720,896		$720,896
Storage Expansion Unit	5057	$5,000	1	$5,000	$10,560	$15,560
Storage Expansion Unit	5058	$5,000	11	$55,000	$116,160	$171,160
1063 Mbps System Unit Expansion Tower	5073	$14,900	1	$14,900		$14,900
1063 Mbps Storage Expansion Tower	5083	$14,900	10	$149,000	$144,000	$293,000
Token Ring Adapter	6149	$1,200	7	$8,400		$8,400
RAID Disk Unit Controller	6533	$9,900	21	$207,900		$207,900
4 GB 1/4-inch Tape Drive	6382	$1,800	1	$1,800		$1,800
RPQ 843804	843804	$900	1	$900		$900
LAN/WAN/Workstation IOP	2629	$2,600	7	$18,200		$18,200
V,24./EIA232 20 ft Cable	0330	$125	1	$125		$125
17.2 GB Disk Unit	6714	$3,800	40	$152,000		$152,000
8.58 GB Disk Unit	6713	$2,100	315	$661,500		$661,500
Base 8.58 GB Disk Unit	8713	$800	1	$800		800
Server Subtotal				*$2,647,117*	*$362,592*	*$3,009,709*
Client Hardware:						
AS/400 9401-150 Growth Server Pkg V4R1	0194	$12,000	1	$12,000	$4,272	$16,272
4.194 GB Disk Unit	Included in S40 base price		1	$0		$0
128 MB Base Memory	Included in S40 base price		1	$0		$0
Twinax Work Station Controller	Included in S40 base price		1	$0		$0
Base Token Ring Adapter (9724)	Included in S40 base price		1	$0		$0
Tape Drive	Included in S40 base price		1	$0		$0
64 MB Additional Main Storage Memory	3110	$768	1	$768		$768
Token Ring Adapter	2724	$840	1	$840		$840
4.194 GB Single Disk Unit	6607	$1,300	1	$1,300		$1,300
Single Client Subtotal				*$14,908*	*$4,272*	*$19,180*
Number of Clients			97			
Client Subtotal				*$1,446,076*	*$414,384*	*$1,860,460*
Server Software:						
IBM Operating System/400 V4R1M0 60-user	Included in S		1	$0		$0
Client Software:						
IBM Operating System/400 V4R1M0	Included in S		97	$0		$0
CICS for OS/400	5716-DFH	$2,450	97	$237,650		$237,650
ILE COBOL	5738-CB1	$900	1	$900		$900
Application Development Toolkit	5716-PW1	$900	1	$900		$900
DB2 Query Manager Dev. Toolkit	Included in S		1	$0		$0
ILE C	5716-CX2	$900	1	$900		$900
Software Subtotal				*$240,350*		*$240,350*
User Connectivity:						
16-port Token Ring Switch (including spare)	8272-216	$9,995	7	$69,967		$69,967
4-port UFC for 8272 (10% spares)	9196	$1,600	12	$19,200		$19,200
TOKEN RING CAU (including spare)	8230-013	$3,340	5	$16,698		$16,698
4-port LIU for 8230 (including spare)	2009	$300	17	$5,100		$5,100
20-PORT TOKEN RING LAM for 8030 (including spar	6738	$1,830	6	$10,981		$10,981
8-port Token Ring connection unit (10% spares)	8226	$545	4915	$2,678,675		$2,678,675
User Connectivity Subtotal				*$2,800,621*		*$2,800,621*
5-year System Subtotal				*$7,134,164*	*$776,976*	*$7,911,140*
Discounts						
Revenue Allowance				($2,104,578)		*($2,104,578)*
Midrange System Option (MSRO)					($132,086)	*($132,086)*
Extended Maintenance Option (EMO)					($109,631)	*($109,631)*
5-year System Total				$5,029,586	$535,259	**$5,564,844**

Notes:

Revenue Allowance is applied to hardware and software for the priced configuration.

EMO is a discount for prepayment of 5 years of maintenace costs.

MSRO is available to customers when agreement is reached for the customer to perform certain duties

Five-Year Cost of Ownership: $5,564,844

tpmC Rating: 43,169.85

$/tpmC: $128.91

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

Numerical Quantities Summary for IBM AS/400e server s40-2208

MQTH, computed Maximum Qualified Throughput	43,169.85
% throughput difference, reported & reproductivity	0.35%

Response Times (in seconds)	90%	Avg.	Max.
- New Order	0.71	0.41	95.20
- Payment	0.55	0.29	95.11
- Order-Status	0.70	0.39	8.72
- Delivery (interactive portion)	0.28	0.15	94.72
- Delivery (deferred portion)	3.24	1.82	19.13
- Stock-Level	1.02	0.49	95.49
- Menu	1.20	0.42	160.58

- Response time delayed for emulated components 0.0

Transaction Mix (in percent of total transactions)	
- New Order	44.34%
- Payment	43.42%
- Order-Status	4.10%
- Delivery	4.06%
- Stock-Level	4.05%

Keying/Think Times (in seconds)	Min.	Avg.	Max.
- New Order	18.00/.01	18.11/122.21	19.98/122.21
- Payment	3.00/.01	3.05/12.21	5.10/122.20
- Order-Status	2.00/.01	2.05/10.25	3.02/102.4
- Delivery	2.00/.01	2.04/5.23	3.18/52.20
Stock-Level	2.00/.01	2.04/5.23	2.96/52.03

Test Duration	
- Ramp-up time	59 minutes
- Measurement interval	20 minutes
- Number of checkpoints	0
- Checkpoint interval	0 minutes
- Number of transactions (all types) completed in Measurement Interval	1,946,868

Transparent file synchronization occurred more than five times during the measurement interval.

Preface

TPC Benchmark™ C Standard Specification was developed by the Transaction Processing Performance Council (TPC) and released August 13, 1992.

This is the full disclosure report for benchmark testing of the IBM AS/400 systems according to the TPC Benchmark“ C Standard Specification. Measurements were done on the AS/400e server s40-2208.

TPC Benchmark™ C exercises the system components necessary to perform tasks associated with that class of on-line transaction processing (OLTP) environments emphasizing a mixture of read-only and update intensive transactions. This is a complex OLTP application environment exercising a breadth of system components associated with such environments characterized by:

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

The benchmark defines four on-line transactions and one deferred transaction, intended to emulate functions that are common to many OLTP applications. However, this benchmark does not reflect the entire range of OLTP requirements. The extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other work loads or environments. Extrapolations to any other environment are not recommended.

Benchmark results are highly dependent upon work load, specific application requirements, 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.

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.

Table of Contents

1.0 General Items

1.1 Application Code Disclosure

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

Appendix D contains the AS/400 application code for the five TPC Benchmark™ C transactions and the terminal functions.

1.2 Benchmark Sponsor

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

This benchmark was sponsored by **International Business Machines** Corporation.

1.3 Parameter Settings

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

- *Database tuning options.*
- *Recovery/commit options.*
- *Consistency/locking options.*
- *Operating system and application configuration parameters.*

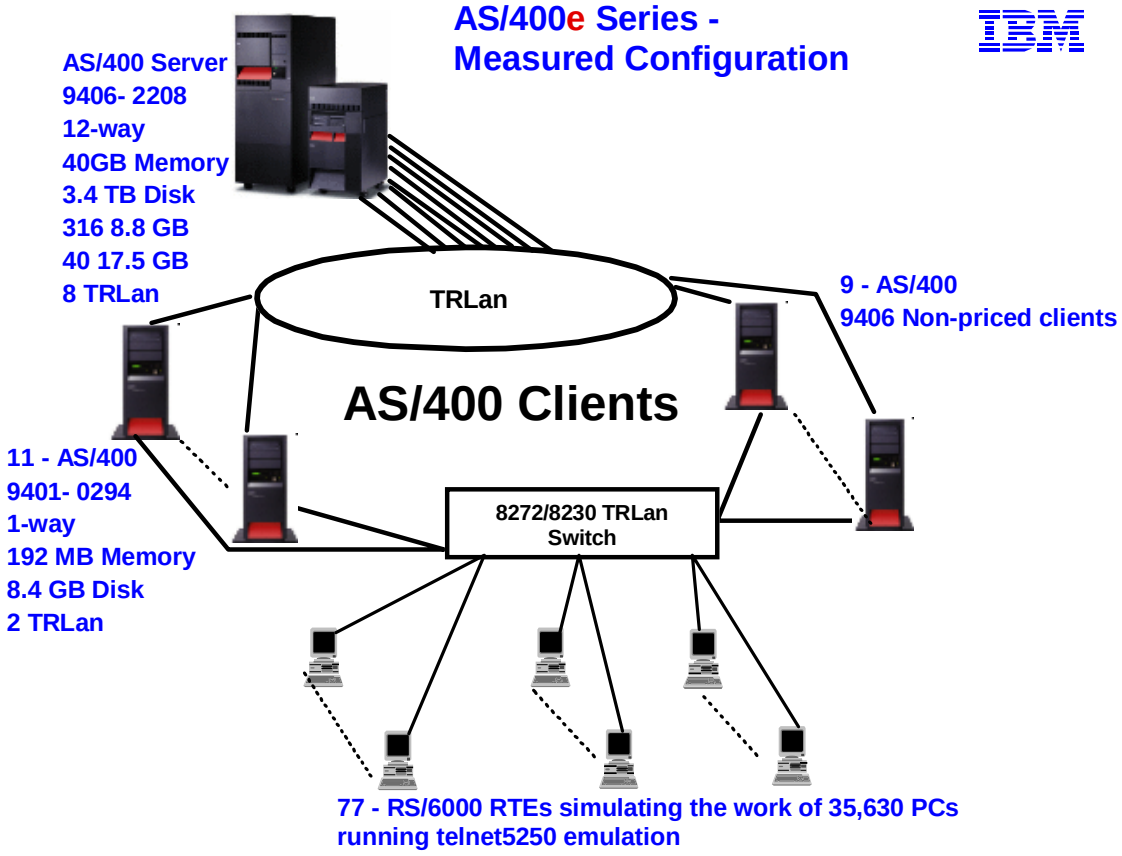
Appendix A contains the system, database, and application parameters changed from their default values used in these TPC Benchmark™ C tests.

1.4 Configuration Diagrams

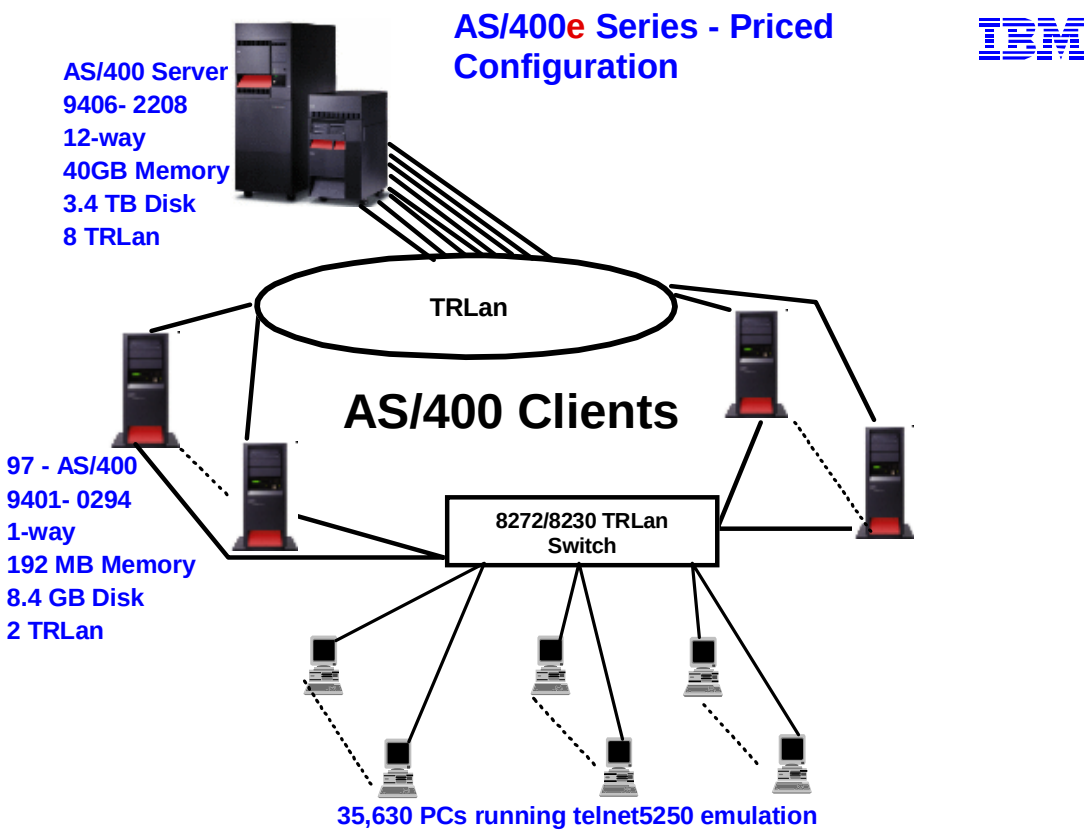
Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

- *Number and type of processors.*
- *Size of allocated memory, and any specific mapping/partitioning of memory unique to the test.*
- *Number and type of disk units (and controllers if applicable).*
- *Number of channels or bus connections to disk units, including the protocol type.*
- *Number of LAN (eg, Ethernet) connections, including routers, workstations, terminals, etc, that were physically used in the test or are incorporated into the pricing structure (see Clause 8.1.8).*
- *Type and run-time execution location of software components (eg, DBMS, client processes, transaction monitors, software drivers, etc).*

1.5 AS/400e server s40-2208 Benchmark Configuration



1.6 AS/400e server s40-2208 C/S Priced Configuration



2.0 Clause 1: Logical Database Design - Related Items

2.1 Table Definitions

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

The listings for all file definitions to create the database files are available in Appendix B and the programs used for loading the database (minimal population) are provided in Appendix C.

2.2 Database Organization

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

Physical space for each file (table) is allocated by OS/400 as the file is filled. Although the initial build and the application transactions add records (rows) to several files in the same transaction, each file's extent will reside in separate areas on physical disk. OS/400 will spread the extents for each file across all available disk units to ensure that multiple access requests to the same file may be handled simultaneously. Records are added contiguously within extents, crossing page boundaries where necessary.

Files are created in sequential order according to their primary key.

Indices are generated concurrently with data for Warehouse, District, and Item tables. All other indices are generated after the database is populated. The available space within the index is included in the space reported in this disclosure.

2.3 Insert and/or Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restriction in the SUT 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.

During the course of the testing, records were inserted into the ITEM file while users were executing the defined TPC-C transactions.

All of the files used by TPC-C transactions were created with the following attributes:

Authority	*PUBLIC	(Any user can view and modify the files).
ALWUPD	*YES	(Allow update and insert of records).
ALWDLT	*YES	(Allow delete of records).
ALWWRT	*YES	(Allow write of records).

Static files were created with *NOMAX specified on the number of records allowed. This limit is therefore set by the operating system at 2,147,483,646 records. Dynamic files were created with an initial size that is slightly larger than initial database requirement, and extent definitions that would allow expansion, as needed.

2.4 Horizontal or Vertical Partitioning

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

For the benchmark test of the AS/400e server s40-2208, IBM did not implement horizontal or vertical partitioning upon any of the benchmark files.

3.0 Clause 2: Transaction and Terminal Profiles - Related Items

3.1 Verification for the Random Number Generator

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

The srandom(), getpid(), and gettimeofday() functions are used to produce unique random seeds for each driver. The drivers use these seeds to seed the srand(), srandom(), and srand48() functions. Random numbers are produced using wrappers around the standard system random number generators.

The negative exponential distribution uses the following function to generate the distribution. This function has the property of producing a negative exponential curve with a specified average and a maximum value 4 times the average.

```
const double RANDOM_4_Z=0.89837799236185
const double RANDOM_4_K=0.97249842407114

double neg_exp_4(double average {
        return - average * (1/RANDOM_4_Z * log (1 - RANDOM_4_K * drand48())));
}
```

The random functions used by the driver system and the database generation program were verified. The C_LAST column was queried to verify the random values produced by the database generation program. After a measurement, the HISTORY, ORDER, and ORDER_LINE tables were queried to verify the randomness of values generated by the driver. The rows were counted and grouped by customer and item numbers.

Here is an example of one SQL query used to verify the random number generation functions:

- create table TEMP (W_ID int, D_ID, C_LAST char(16), CNTR int);
- insert into TEMP select C_W_ID, C_D_ID, C_LAST, COUNT(*) from CUSTOMER group by C_W_ID, C_D_ID, C_LAST;
- select CNTR, COUNT(*) from TEMP group by CNTR order by 1;

3.2 Input/Output Screens

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

The Data Description Specification for the I/O screens are available in Appendix D--Application Source Code. The screens are defined as required in the TPC Benchmark™ C specification except for the "total amount" field of the New-Order screen. The "total amount" field has been shifted one column to the left to allow space for the attribute byte for the menu lines in the next row. This is a limitation of the implementation of the screen I/O. IBM did not gain any performance improvement from this change.

3.3 Terminal Features

The method used to verify that the terminals configured provide all the features described in Clause 2.2.2.4 must be disclosed.

The following numbered items correspond directly to the seven items listed under Clause 2.2.2.4 with a description of how the requirement was met.

1. *The input characters appear on the input/output screen as they are keyed in:*

The RTE used in these tests (see Appendix E for a detailed description) emulates only the display attached to the SUT via a token-ring LAN. The 5250 workstations being emulated support local echo of characters as they are keyed.

2. *Input is allowed only in the positions of an input field:*

> Data Description Specification (DDS) is used to define the areas of the screen where input is allowed as input/output capable. Other fields designated as output only, as labels, or as blank are protected by locking the keyboard from entry.

3. *Input-capable fields are designated by some method of clearly identifying them:*

> The input-capable fields of this benchmark are distinguishable by an underscore character the length of the field and is specified by using the attribute for field underscore in DDS.

4. *It must be possible to key in only significant characters into fields:*

> The fields for the transactions have been specified for translating nonkeyed positions in a field to either blanks or zeroes for alphanumeric, or zeroes for numeric fields through DDS keywords. Therefore, if the Warehouse field is specified as having four positions and the actual warehouse used is "1," then the user only need enter "1" and DDS will fill in the field with zeroes for the nonkeyed positions and right-justify the field so that the actual field input would be "0001."

5. *All fields for whicha value is necessary to allow the application to complete are required to contain input prior to the start of the measurement of the transaction RT, or the application must contain a set of error-handling routines to inform the user that required fields have not been entered:*

> The application code for each of the transactions is written to require the minimum input required for completion, if the minimum is not input then the application displays an error message and returns for more input.

6. *Fields can be keyed and rekeyed in any order: Specifically.*

> *The emulated user must be able to move the input cursor forward and backward directly to the input capable fields.*
> *The application cannot rely on fields being entered in any particular order.*
> *The user can return to a field that has been keyed in and change its value prior to the start of the measurement of the transaction RT.*
>
> The screen handling properties defined in DDS are resident in the AS/400 workstation I/O processor. All fields are first entered and verified by the I/O processor and then sent to the CPU at the start of the transaction. The I/O processor does not require that the fields be entered in any order, and the fields are sent to the CPU when the "ENTER" key is pressed.

7. Numeric fields must be protected from nonnumeric input:

AS/400 DDS allows specification of numeric-only fields and if a nonnumeric character is entered, a message is sent to the display and the keyboard is locked.

3.4 Presentation Managers

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

The workstation emulated in the measured configuration is an IBM 5250-type workstation. The 5250 workstations do not use a presentation manager nor is it an intelligent workstation. All of the processing of the input/output screens is handled by the SUT.

Workstations used in the priced configuration are PCs emulating 5250 workstations. The PCs are connected to the AS/400 via 16 Mb token-ring LAN.

3.5 Home and Remote Order Lines

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

Table 1 on page 10 shows the percentage of home and remote transactions that occurred during the measurement period for the New-Order transactions.

3.6 New-Order Rollback transactions

The percentage of New-Order transactions that were rolled back as a result of an illegal item number must be disclosed.

Table 1 on page 10 shows the percentage of New-Order transactions that were rolled back due to an illegal item being entered.

3.7 Number of Items per Order

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

Table 1 on page 10 shows the average number of items ordered per New-Order transaction.

3.8 Home and Remote Payment Transactions

The percentage of Home and Remote Payment transactions must be disclosed.

Table 1 on page 10 shows the percentage of Home and Remote transactions that occurred during the measurement period for the Payment transactions.

3.9 Nonprimary Key Transactions

The percentage of Payment and Order-Status transactions that used nonprimary key (C_LAST) access to the database must be disclosed.

Table 1 on page 10 shows the percentage of nonprimary key access to the database by the Payment and Order-Status transactions.

3.10 Skipped Delivery Transactions

The percentage of Delivery transactions that were skipped as a result of an insufficient number of rows in the NEW-ORDER table must be disclosed.

Table 1 on page 10 shows the percentage of Delivery transactions missed due to a shortage of supply in the NEW-ORDER table.

3.11 Mix of Transaction Types

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

Table 1 on page 10 shows the mix percentage for each of the transaction types executed by the SUT.

3.12 Queueing Mechanism of Delivery

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

Deferred queuing of the Delivery transaction is handled within standard support from the CICS transaction monitor.

Table 1. Numerical Quantities for Transaction and Terminal Profiles	
New Order	AS/400e server s40-2208
Percentage of Home order lines	99.01%
Percentage of Remote order lines	0.99%
Rolled Back Transactions	1.00%
Number of Items per order	10
Payment	
Percentage of Home transactions	85.04%
Percentage of Remote transactions	14.96%
Nonprimary Key Access	
Percentage of Payment using C_LAST	60.02%
Percentage of Order-Status using C_LAST	60.01%
Delivery	
Delivery transactions skipped	0
Transaction Mix	
New-Order	44.34%
Payment	43.42%
Order-Status	4.10%
Stock-Level	4.06%
Delivery	4.05%

4.0 Clause 3: Transaction and System Properties - Related Items

The results of the ACID test must be disclosed along with a description of how the ACID requirements were met.

4.1 Atomicity Requirements

The system under test must guarantee that 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.

4.1.1 Atomicity of Completed Transaction

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

The following steps were performed to verify the atomicity of completed transactions:

1. Randomly selecta Customer, District, and Warehouse, and query the Customer, District, and Warehouse tables.
2. Execute a Payment transaction for the Customer, District, and Warehouse used in Step1 above. Commit the transaction.
3. Repeat the query performed in Step1 to demonstrate that the appropriate changes have been made.

4.1.2 Atomicity of Aborted Transactions

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

The following steps were performed to verify the atomicity of the aborted Payment transaction:

1. Randomly selecta Customer, District, and Warehouse, and query the Customer, District, and Warehouse tables.
2. Execute a Payment transaction for the Customer, District, and Warehouse used in Step1 above. Roll-back the transaction.
3. Repeat the query performed in Step1 to verify that no changes have been made to the database.

4.2 Consistency Requirements

Consistency is the property of the application that requires an 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.

4.2.1 Consistency Condition 1

Entries in the WAREHOUSE and DISTRICT tables must satisfy the relationship:

W_YTD = sum(D_YTD)

for each warehouse defined by (W_ID = D_W_ID)

The following SQL queries were executed before and after transactions were run to show that the database was always in a consistent state.

```
Select WID, WYTD from WRHS
Select DWID, sum(DYTD) from DSTRCT group by DWID
```

The results of these two queries were then compared to verify consistency.

4.2.2 Consistency Condition 2

Entries in the DISTRICT, ORDER, and NEW-ORDER tables must satisfy the relationship:

D_NEXT_O_ID -1 = max(O_ID) = max(NO_O_ID)

for each district defined by (D_W_ID = O_W_ID) = NO_W_ID) and (D_ID = O_D_ID = NO_D_ID). This condition does not apply to the NEW-ORDER table for any districts which have no outstanding new orders.

The following SQL queries were executed before and after transactions were run to show that the database was always in a consistent state.

```
Create View QRYTEMP1(OWID, ODID, MAXOID)
  As Select OWID, ODID, MAX(OID) from ORDERS group by OWID, ODID
Create View QRYTEMP2(NOWID, NODID, MAXNOOID)
  As Select NOWID, NODID, MAX(NOOID) from NEWORDS group by NOWID, NODID
Select DWID, DID, (DNXTOR-1), MAXOID, MAXNOOID
  from DSTRCT, QRYTEMP1, QRYTEMP2 where DWID = OWID and DWID=NOWID
  and DID =ODID and DID = NODID and (((DNXTOR-1) <> MAXOID)
  or ((DNXTOR-1) <> MAXNOOID) or (MAXOID <> MAXNOOID))
```

If any records are produced by these queries, the database would be inconsistent. No records were produced by these queries.

4.2.3 Consistency Condition 3

Entries in NEW-ORDER table must satisfy the relationship:

max(NO_O_ID) - min(NO_O_ID) + 1 = {number of rows in the NEW-ORDER table for this district}

for each district defined by NO_W_ID and NO_D_ID. This condition does not apply to any districts which have no outstanding new orders.

The following SQL queries were executed before and after transactions were run to show that the database was always in a consistent state.

```
Create View QRYTEMP3(NOWID, NODID, MAXNOOID, MINNOOID, MAXMIN, COUNTO ID)
  As Select NOWID, NODID, MAX(NOOID), MIN(NOOID), (MAX(NOOID) -
  MIN(NOOID) +1), COUNT(*) from NEWORD group by NOWID, NODID
Select * from QRYTEMP3 where MAXMIN <> COUNTOID
```

If any records are produced by these queries, the database would be inconsistent. No records were produced by these queries.

4.2.4 Consistency Condition 4

Entries in the ORDER and ORDER-LINE tables must satisfy the relationship:

sum(O_OL_CNT) = {number of rows in the ORDER-LINE table for this district}

for each district defined by (O_W_ID = OL_W_ID) and (O_D_ID = OL_D_ID).

The following SQL queries were executed before and after transactions were run to show that the database was always in a consistent state.

```
Create View QRYTEMP4(OLWID, OLDID, COUNTORD)
  As Select OLWID, OLDID, COUNT(*)
  From ORDERLINE Group by OLWID, OLDID
Create View QRYTEMP5(OWID, ODID, SUMOLINES)
  As Select OWID, ODID, SUM(OLINES)
  From ORDERS Group by OWID, ODID
Select OWID, ODID, SUMOLINES, COUNTORD
  From QRYTEMP4, QRYTEMP5 Where OWID = OLWID and ODID = OLDID
  and SUMOLINES <> COUNTORD Order by OWID, ODID
```

If any records are produced by these queries, the database would be inconsistent. No records were produced by these queries.

4.2.5 Consistency Condition 5

For any row in the ORDER table, O_CARRIER_ID is set to a null value if and only if there is a corresponding row in the NEW-ORDER table defined by (O_W_ID, O_D_ID, O_ID) = (NO_W_ID, NO_D_ID, NO_O_ID).

This consistency test completed successfully.

4.2.6 Consistency Condition 6

For any row in the ORDER table, O_OL_CNT must equal the number of rows in the ORDER-LINE table for the corresponding order defined by (O_W_ID, O_D_ID, O_ID) = (OL_W_ID, OL_D_ID, OL_O_ID).

This consistency test completed successfully.

4.2.7 Consistency Condition 7

For any row in the ORDER-LINE table, OL_DELIVERY_D is set to a null date/time if and only if the corresponding row in the ORDER table defined by (O_W_ID, O_D_ID, O_ID) = (OL_W_ID, OL_D_ID, OL_O_ID) has (O_CARRIER_ID) set to a null value.

This consistency test completed successfully.

4.2.8 Consistency Condition 8

Entries in the WAREHOUSE and HISTORY tables must satisfy the relationship:

W_YTD = sum(H_AMOUNT)

for each warehouse defined by (W_ID = H_W_ID).

This consistency test completed successfully.

4.2.9 Consistency Condition 9

Entries in the DISTRICT and HISTORY tables must satisfy the relationship:

D_YTD = sum(H_AMOUNT)

for each district defined by (D_W_ID, D_ID = H_W_ID, H_D_ID).

This consistency test completed successfully.

4.2.10 Consistency Condition 10

Entries in the CUSTOMER, HISTORY, ORDER, and ORDER-LINE tables must satisfy the relationship:

C_BALANCE = sum(OL_AMOUNT) - sum(H_AMOUNT)
where:
H_AMOUNT is selected by (C_W_ID, C_D_ID, C_ID) = (H_C_W_ID, H_C_D_ID, H_C_ID)
and:
OL_AMOUNT is selected by:
(OL_W_ID, OL_D_ID, OL_O_ID) = (O_W_ID, O_D_ID, O_ID) and
(O_W_ID, O_D_ID, O_C_ID) = (C_W_ID, C_D_ID, C_ID) and
(OL_DELIVERY_D is not a null value)

This consistency test completed successfully.

4.2.11 Consistency Condition 11

Entries in the CUSTOMER, ORDER, and NEW-ORDER tables must satisfy the relationship:

(count(*) from ORDER) - (count(*)from NEW-ORDER) = sum(C_DELIVERY_CNT)

for each district defined by (O_W_ID, O_D_ID) = (NO_W_ID, NO_D_ID)= (C_W_ID, C_D_ID).

This consistency test completed successfully.

4.2.12 Consistency Condition 12

Entries in the CUSTOMER, and ORDER-LINE table must satisfy the relationship:

C_BALANCE + C_YTD_PAYMENT = sum(OL_AMOUNT)

for any randomly selected customers and where OL_DELIVERY_ID is not set to a null date/time.

This consistency test completed successfully.

All 12 consistency tests were completed successfully.

4.2.13 Consistency Tests

Verify that the database is initially consistent by verifying that it meets the consistency conditions defined in Clauses 3.3.2.1 to 3.3.2.4. Describe the steps used to do this in sufficient detail so that the steps are independently repeatable.

The queries defined in 4.2.1 through 4.2.4 were run after initial database build and prior to executing any transactions. All queries showed that the database was in a consistent state.

After executing transactions at full load for approximately 10 minutes, the queries defined in 4.2.1 through 4.2.4 were run again. All queries showed that the database was still in a consistent state.

4.3 Isolation Requirements

Operations of concurrent database transactions must yield results which are indistinguishable from the results which would be obtained by forcing each transaction to be serially executed to completion in some order.

4.3.1 Isolation Test 1

This test demonstrates isolation for read-write conflicts of Order-Status and New-Order transactions.

The following steps were performed to satisfy the test of isolation for Order-Status and New-Order transactions:

1. First terminal: Starta New-Order transaction with the necessary inputs. The transaction is delayed to pause the program execution.

2. Second terminal: Start an Order-Status transaction for the same customer as used in the New-Order transaction.
3. Second terminal: The Order-Status transaction attempts to read the CUSTOMER file but is locked out by the New-Order transaction waiting to complete.
4. First terminal: The New-Order transaction is released and the Commit is executed releasing the record. With the CUSTOMER record now released, the Order-Status transaction can now complete.
5. Second terminal: Verify that the Order-Status transaction completes after the New-Order transaction. and that the results displayed for the Order-Status transaction match the input for the New-Order transaction.

4.3.2 Isolation Test 2

This test demonstrates isolation for read-write conflicts of Order-Status and New-Order transactions when the New-Order transaction is rolled back.

The following steps were performed to satisfy the test of isolation for Order-Status and a rolled back New-Order transaction.

1. First terminal: Performa New-Order transaction for the same customer in the Order-Status transaction in Isolation Test 1; include an invalid item number in the order. The transaction is delayed just prior to the rollback.
2. Second terminal: Start an Order-Status transaction for the same customer used in the the New-Order transaction. The Order-Status transaction attempts to read the CUSTOMER file but is locked by the New-Order transaction.
3. First terminal: Roll back the New-Order transaction. With the CUSTOMER record now released, the Order-Status transaction completes.
4. Verify the results from the Order-Status transaction matches those in Isolation Test 1.

4.3.3 Isolation Test 3

This test demonstrates isolation for write-write conflicts of two New-Order transactions.

1. The following steps were performed to verify isolation of two New-Order transactions:
2. First terminal: Starta New-Order transaction using the necessary inputs. The transaction is delayed just prior to the Commit.
3. Second terminal: Starta second New-Order transaction for the same customer used by the first terminal. This transaction is forced to wait while the first terminal holds a lock on the DISTRICT record requested by the second terminal.
4. First terminal: The New-Order transaction is allowed to complete and Commit the transaction. With the DISTRICT record released, the second terminal New-Order transaction will complete.
5. Verify the order number from the second terminal New-Order transaction is one greater than the order number from the first terminal.

4.3.4 Isolation Test 4

This test demonstrates isolation for write-write conflicts of two New-Order transactions when one transaction is rolled back.

The following steps were performed to verify isolation of two New-Order transactions after one is rolled back:

1. First terminal: Starta New-Order transaction using the necessary inputs to causea roll back (invalid item number). The transaction is delayed just prior to the rollback.
2. Second terminal: Starta second New-Order transaction for the same customer used by the first terminal. This transaction is forced to wait while the first terminal holds a lock on the DISTRICT record requested by the second terminal.
3. First terminal: The New-Order transaction is allowed to complete, and the transaction is rolled back due to the invalid item number.
4. Second terminal: With the DISTRICT record released, the second terminal New-Order transaction will complete normally.
5. Verify the order number from the second terminal New-Order transaction is equal to the next order number before either New-Order transaction was started.

4.3.5 Isolation Test 5

This test demonstrates isolation for write-write conflicts of Payment and Delivery transactions.

The following steps were performed to successfully conduct this test:

1. First terminal:A Delivery transaction is started. The transaction is delayed just prior to the Commit.
2. Second terminal: Starta Payment transaction fora customer that will have an order delivered in this transaction started in Step 1. The Payment transaction is forced to wait while the Delivery transaction holds a lock on the CUSTOMER record.
3. The Delivery transaction completes.
4. Second terminal: With the CUSTOMER record released, the Payment transaction is now able to complete.

4.3.6 Isolation Test 6

This test demonstrates isolation for write-write conflicts of Payment and Delivery transactions when the Delivery transaction is rolled back.

The following steps were performed to successfully conduct this test:

1. First terminal: Starta Delivery transaction. The transaction is delayed just prior to the rollback.
2. Second terminal: Starta Payment transaction fora customer that will have an order delivered in this transaction started in Step 1. The Payment transaction is forced to wait while the Delivery transaction holds a lock on the CUSTOMER record.
3. The Delivery transaction rolls back.
4. Second terminal: With the CUSTOMER record released, the Payment transaction is now able to complete.

4.3.7 Isolation Test 7

This test demonstrates repeatable reads for the New-Order transaction while an interactive transaction updates the price of an item.

The following steps were performed to successfully conduct this test:

1. First terminal: Executea New Order transaction including itemsx and y.

2. First terminal:A New-Order transaction is started that contains itemx twice and itemy once. This transaction is stopped after reading the price of item x from the item file the first time.
3. Second terminal: Using interactive SQL, an update transaction is started for itemsx and y, increasing their price by 10%. Case A, transaction 3 stalls, occurs.
4. First terminal: The New-Order transaction is allowed to complete. It is verified that the prices for items x and y are the same throughout the entire transaction and that they match the results of Step 1.
5. Second terminal: After the New-Order transaction completes, the update transaction completes and is committed.
6. First terminal: Step1 is repeated, noting that the prices of itemsx andy now match those set in Step 3.

4.3.7.1 Isolation Test 8

This test demonstrates isolation for phantom protection between a Delivery and a New-Order transaction.

The following steps were performed to successfully conduct this test:

1. First terminal: All rows for a randomly selected district and warehouse were removed from the NEW-ORDER table.
2. First terminal:A Delivery transaction for the selected warehouse was started.
3. First terminal: The Delivery transaction was stopped immediately after reading the NEW-ORDER table for the selected district. No qualifying row was found.
4. Second terminal:A New-Order transaction was started for the same warehouse and district. Case A, Transaction 2 stalled.
5. First terminal: Repeated the read of the NEW-ORDER table for the selected district.
6. Again no qualifying row was found.
7. First terminal: The Delivery transaction was allowed to complete and was COMMITTED.
8. Second terminal: The NEW-ORDER transaction completed successfully.

4.3.7.2 Isolation Test 9

This test demonstrates isolation for phantom protection between an Order-Status and a New-Order transaction.

The following steps were performed to successfully conduct this test:

1. First terminal: An Order-Status transaction for a selected customer was started.
2. First terminal: The Order-Status transaction was stopped immediately after reading the ORDER table for the selected customer. The most recent order for that customer was found.
3. Second terminal:A NEW-ORDER transaction was started for the same customer. Case A, Transaction 2 stalled.
4. First terminal: Repeated the read of the ORDER table for the selected customer.
5. Verified the order found was the same as in step 3.
6. First terminal: The Order-Status transaction was allowed to complete and was COMMITTED.
7. Second terminal: The NEW-ORDER transaction completed successfully.

4.4 Durability Requirements

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

4.4.1 Permanent Unrecoverable Failure of any Single Durable Medium

Permanent unrecoverable failure of any single durable medium containing TPC-C database tables or recovery log data.

The AS/400 implementation of the TPC Benchmark™ C divides the configured disk space into two available auxiliary storage pools (ASP): System ASP and User ASP. The system ASP contains the operating system, integrated relational database, the application libraries, and the TPC-C tables. The User ASP is protected by device parity protection (RAID-5) and contains the journal receiver (recovery log).

4.4.1.1 Failure of Durable Medium of Journal Receiver and Instantaneous Interruption and Memory Failure

The following steps were performed to successfully complete the test of the Durability of the journal receiver:

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving SUM_1.
2. A full scale test was started on the SUT. The test was allowed to run for 10 minutes before creating the failure.
3. The signal cable froma single disk unit in the User ASP was disconnected. Since the User ASP is protected by device parity protection (RAID-5) the system continued to process transactions. Detection of the device failure caused a diagnostic message to be issued. Processing of transactions continued without performance degradation.
4. Processing was allowed to continue for an additional 10 minutes.
5. An instantaneous power failure was then simulated by issuing an immediate power-off at the service panel.
6. The system was then powered on and IPLed.
7. The number of New-Order transactions executed by the SUT is verified against the number of successful transactions logged by the RTE.
8. Step1 above was performed again retrieving the new total of orders processed, SUM_2. The difference between SUM_2 and SUM_1 was compared to the number of transactions reported by RTE.

4.4.1.2 Failure of Durable Medium of Database

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

1. The current count of the total number of orders was determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table giving SUM_1.
2. The DSTRCT database table was saved.
3. A full scale test was started on the SUT. The test was allowed to run for 10 minutes.
4. An unprotected disk in the System ASP was disconnected. This caused the system to halt.
5. The disk was reinstalled and the system restarted.

6. Restore the selected table saved in Step 2. This is equivalent to having losta disk unit and having to restore the system from a backup tape.
7. Step1 is performed returning SUM_2. SUM_2 is equal to SUM_1 returned in Step 1.
8. The OS/400 command APYJRNCHG (Apply Journal Changes) is executed applying all of the changes to the restored table for transactions that occurred after the table was saved.
9. Step1 is performed returning SUM_3.
10. The difference between SUM_3 and SUM_1 is verified to match the total number of successful New-Order transactions completed during the measurement.

5.0 Clause 4: Scaling and Database Population - Related Items

5.1 Cardinality of Tables

The cardinality (ie, the number of rows) of each table, as it existed at the start of the benchmark run, must be disclosed.

Table 2 portrays the TPC Benchmark™ C defined tables and the number of rows for each table as they were built initially.

Table 2. Initial Database Build (# of rows per table)	
TPC Benchmark C Tables	AS/400e server s40-2208
WAREHOUSE	3,563
CUSTOMER	111,000,000
NEW-ORDER	33,300,000
DISTRICT	37,000
STOCK	370,000,000
ORDERS	111,000,000
ORDER-LINE	1,109,993,862
HISTORY	111,000,000
ITEM	100,000

5.2 Distribution of Tables and Logs

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

The IBM AS/400 utilizes a Single-Level Storage concept where OS/400 views all drives in an Auxiliary Storage Pool (ASP) as a single virtual drive. This technique spreads information across all available drives in an ASP, attempting to maintain equivalent percentages of free storage. For this Benchmark, a 316-Disk ASP was used for system code, application code, and the database. A separate, fully mirrored, 40-Disk ASP was used for log data.

5.3 Database Model Implemented

A statement must be provided that describes the database model implemented by the DBMS used.

The type of database implemented in all IBM AS/400 systems is an integrated relational database. The database is integrated into the OS/400 operating system.

5.4 Partitions/Replications Mapping

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

IBM did not implement horizontal or vertical partitioning for these TPC-C tests beyond the normal transparent system partitioning.

6.0 Clause 5: Performance Metrics and Response Time - Related Items

6.1 Response Times

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

Table 3 lists the response times and the ninetieth percentiles for each of the transaction types for the AS/400e server s40-2208.

6.2 Keying and Think Times

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

Table 3 lists the keying and think times from the measured TPC-C tests for the AS/400e server s40-2208.

Table 3. AS/400e server s40-2208 Response, Keying, and Think Times

Response Times	NewOrder	Payment	Order Status	Delivery (int/def)	Stock Level	Menus
90%	0.71	0.55	0.70	0.28/3.24	1.02	1.20
Average	0.41	0.29	0.39	0.15/1.82	0.49	0.42
Maximum	95.20	95.11	8.72	94.72/19.13	95.49	160.58
			Think Times			
Minimum	0.01	0.01	0.01	0.01	0.01	N/A
Average	12.23	12.21	10.25	5.23	5.23	N/A
Maximum	122.21	122.20	102.04	52.20	52.03	N/A
			Keying Times			
Minimum	18.00	3.00	2.00	2.00	2.00	N/A
Average	18.11	3.05	2.05	2.04	2.04	N/A
Maximum	19.98	5.10	3.02	3.18	2.96	N/A

6.3 Client Substitution Table

The following table lists the new orders/user by each priced and non-priced clients, and also the average new orders/users for the set of priced clients and for the set of non-priced (substitute) clients.

Client Substitution table

System	CPU	Users	tpmC	Measurement New Orders	New Orders / user
			Priced Clients		
A	150-0294	370	459.90	9,198	24.86
B	150-0294	370	442.65	8,853	23.93
C	150-0294	370	459.60	9,192	24.84
D	150-0294	370	460.00	9,200	24.86
E	150-0294	370	453.00	9,060	24.49
F	150-0294	370	459.90	9,198	24.86
G	150-0294	370	462.65	9,245	24.97
H	150-0294	370	464.70	9,294	25.12
I	150-0294	370	462.55	9,251	25.00
J	150-0294	370	459.15	9,183	24.82
K	150-0294	370	460.30	9,206	24.88
				Average --	24.79
			Non-Priced Clients		
L	20s	1,780	2095.20	41,904	23..54
M	30s	4,000	4789.15	95,783	23.95
N	170	4,000	4915.55	98,311	24.58
O	170	1,780	2090.15	41,801	23.48
P	170	4,000	4849.00	96,980	24.25
Q	170	4,000	4807.55	96,151	24.04
R	170	4,000	4861.90	97,238	24.31
S	170	4,000	4778.60	95,572	23.89
T	170	4,000	4943.65	98,873	24.72
		35,630		Average --	24.16

6.4 Response Time Frequency Distribution

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

Note: Transaction counts are shown at the lower end of the interval ie, a count of 200 at 0 indicates that there were 200 transactions with a response time between 0 and the next interval.

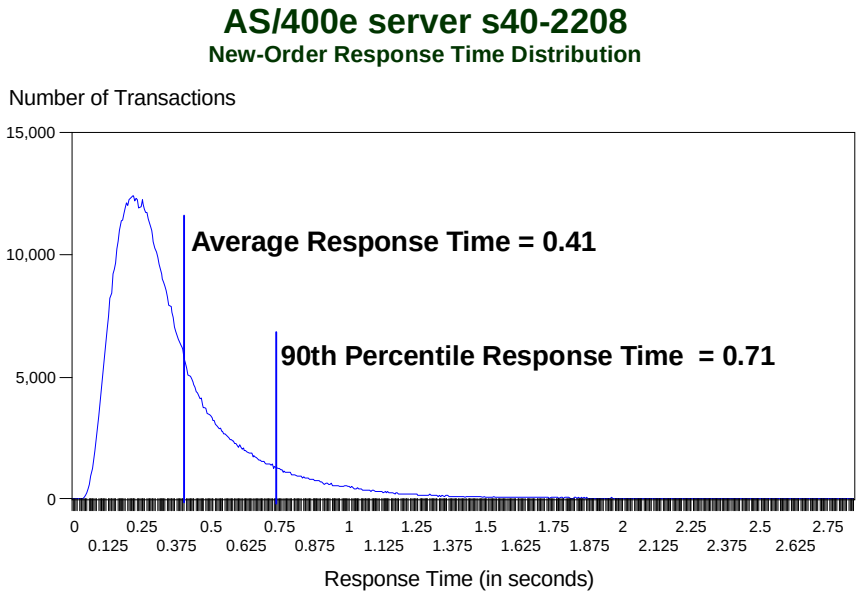


Figure 1. AS/400e server s40-2208 New-Order Response Time Distribution

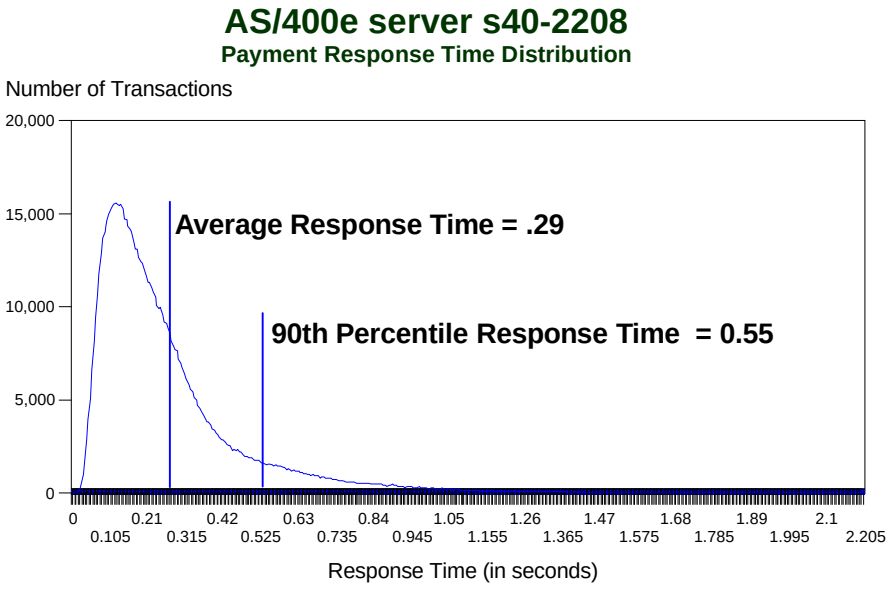


Figure 2. AS/400e server s40-2208 Payment Response Time Distribution

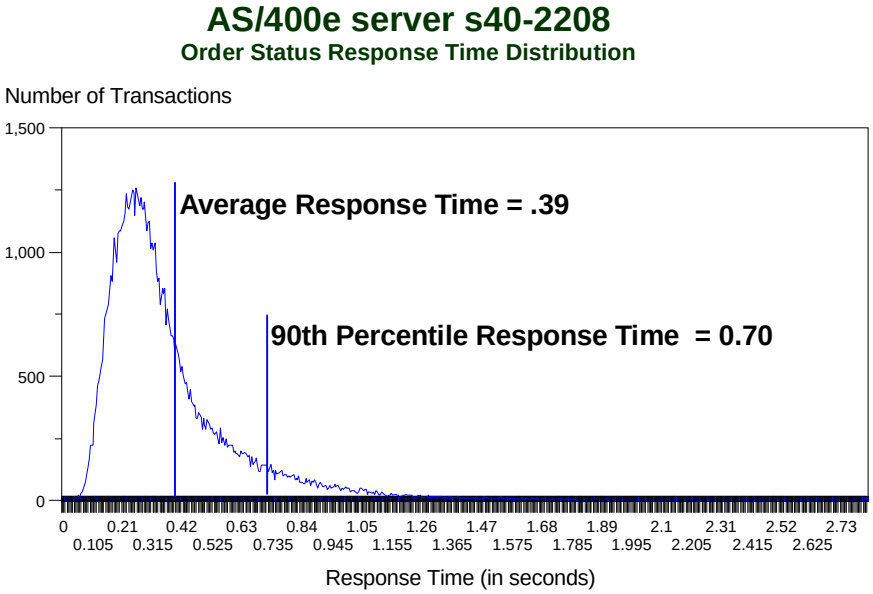


Figure 3. AS/400e server s40-2208 Order-Status Response Time Distribution

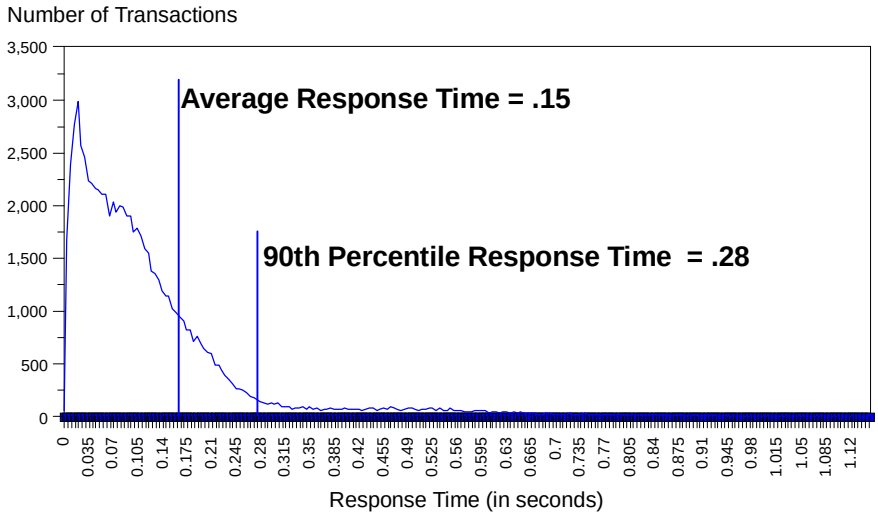


Figure 4. AS/400e server s40-2208 Delivery (Interactive) Response Time Distribution

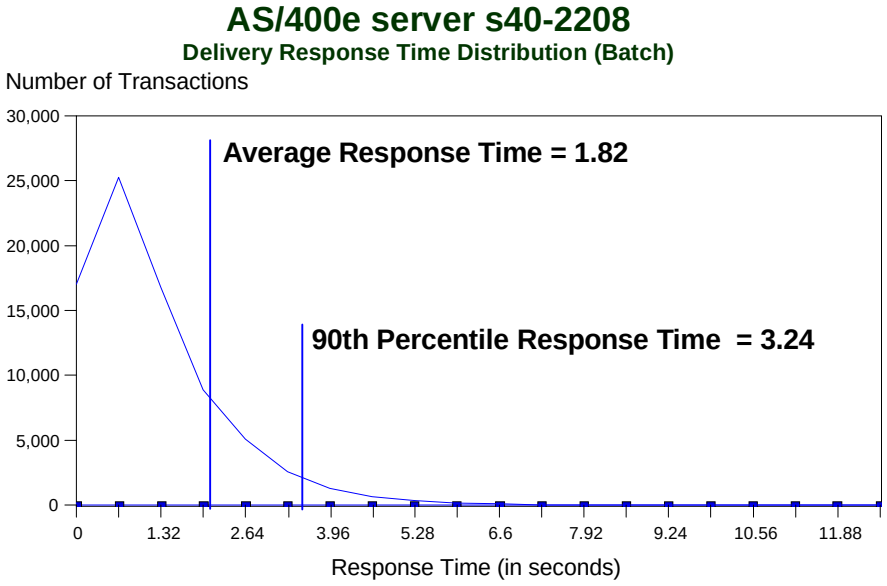


Figure 5. AS/400e server s40-2208 Delivery (Batch) Response Time Distribution

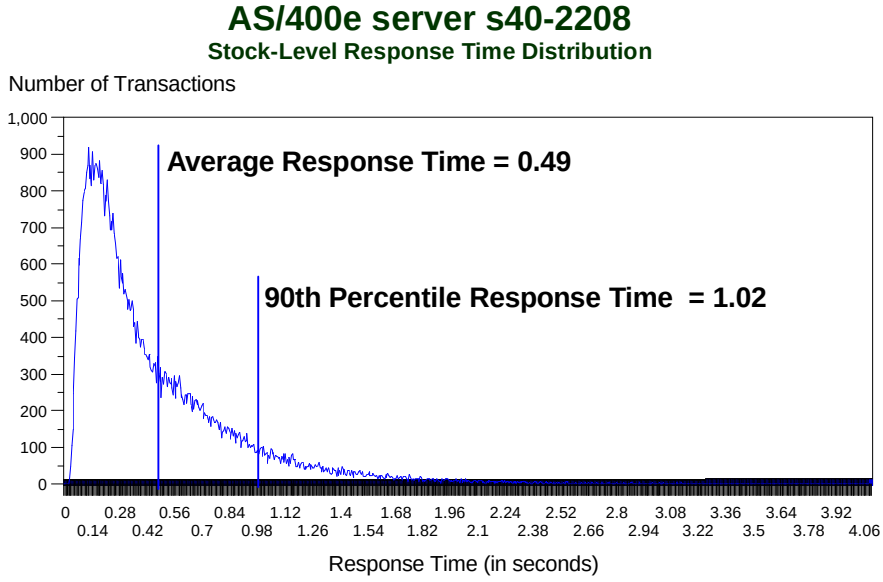


Figure 6. AS/400e server s40-2208 Stock-Level Response Time Distribution

6.5 Performance Curve for Response Time versus Throughput

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

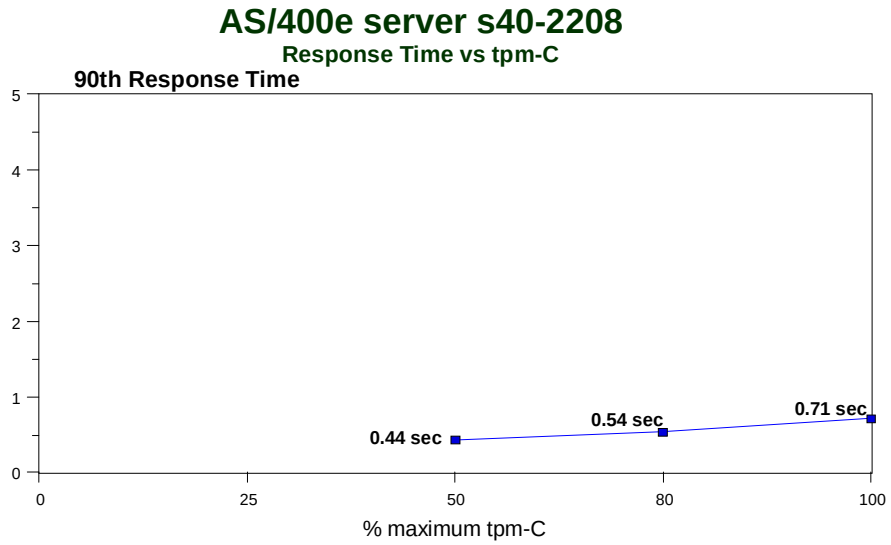


Figure 7. AS/400e server s40-2208 New-Order Response Time Versus Throughput

6.6 Think Time Frequency Distribution

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

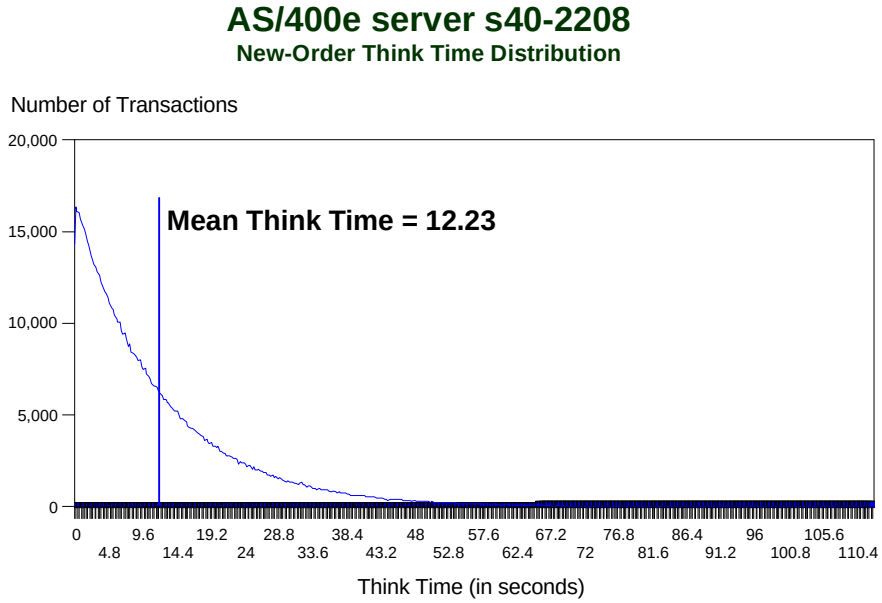


Figure 8. AS/400e server s40-2208 New-Order Think Time Distribution

6.7 Throughput Versus Elapsed Time

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

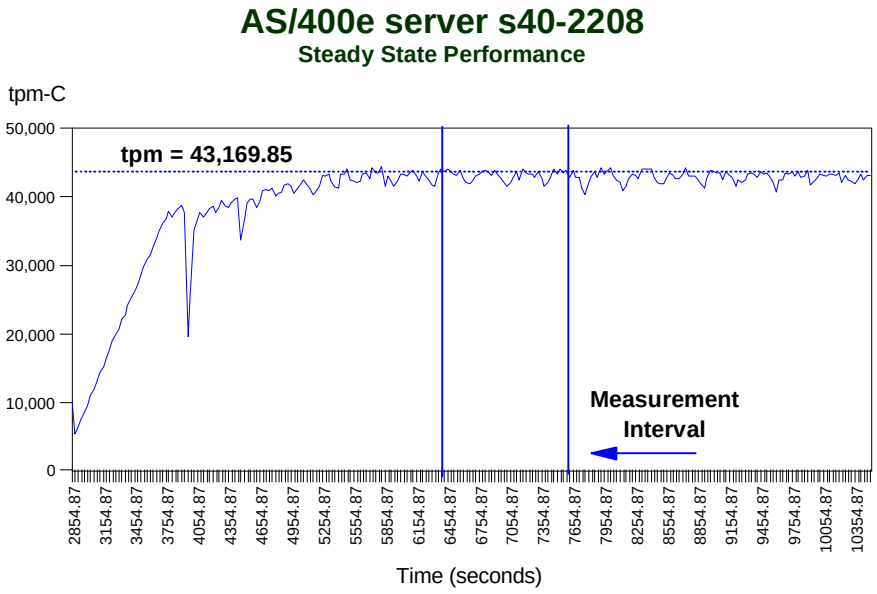


Figure 9. AS/400e server s40-2208 New-Order Throughput Versus Elapsed Time

6.8 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.

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

- At the database transaction start, a “start of commit cycle” entry is recorded in the journal.
- For each of the files updated by a transaction:
  - The database record being updated is locked by the transaction preventing further updates or reads until the journal records are written.
  - A before image of the record is written to the journal receiver.
  - An after image of the record is written to the journal receiver.
- At the end of the database transaction, the records are committed in the journal and all locks on the database records by the transaction are released.

Recording a block of journal entries does not correspond directly to a disk write, since the AS/400 journal management function has the ability to block journal writes from one or more jobs into a single physical I/O.

The AS/400 integrated relational database does not require an overt checkpointing system to ensure that data is written to disk. AS/400 standard journal management function ensures that data in disk files is synchronized with that in memory in a transparent, nondisruptive fashion.

As database I/Os are committed, the journal entries associated with the change are written prior to the completion of the commitment function. The database I/Os are issued as asynchronous I/Os which may be delayed by other requests to use the same data. To ensure that all data updates are completed in a reasonable period of time, AS/400 journal management ensures that unwritten pages from all tables being journaled are forced to disk at least once for every 50,000 entries to the journal. That is, for the 9 tables in TPC-C one file is forced to disk every 5,555 journal entries, so that all 9 are synchronized within the period of time it takes to log 50,000 journal entries.

The AS/400e server s40-2208 system operating at 43,169.85 tpm-C completes 50,000 entries to the journal in approximately 0.86 seconds. A minimum measurement interval of 20 minutes includes approximately 1036 complete cycles.

6.9 Reproducibility

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

A repeatability measurement was taken on the AS/400e server s40-2208 for the same length of time as the measured run. The repeatability interval was taken from the same measurement as the reported interval and the intervals were separated by 12 minutes. The repeatability measurement was 43,018.75tpmC.

6.10 Measurement Interval

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

The measurement interval for the tpmC reported was for 20 minutes in duration. The reproducibility measurement was also for a 20-minute interval.

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

7.1 RTE Availability

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

Appendix E contains the scripts used in the Remote Terminal Emulator testing.

7.2 Functionality and Performance of Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to that of the priced system.

In the benchmark configuration the Remote Terminal Emulator (RTE) communicates with the client system over token ring. The communications mechanism used in the benchmarked and priced configurations are the same.

7.3 Network Bandwidth

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

All networks used 16 Mb/second token ring. The tested configuration of front-end AS/400s and server AS/400 was physically connected on one 16 Mb/second token ring network.

The priced configuration observed the physical limitation of 260 physical devices connected to a single token ring.

7.4 Operator Intervention

If the configuration requires operator intervention, the mechanism and the frequency of this intervention must be disclosed.

The AS/400 configurations reported do not require any operator intervention to sustain the reported throughput during the 8-hour period.

8.0 Clause 7: Pricing - Related Items

8.1 Hardware and Programs Used

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

The detailed list of all hardware and programs for the priced configuration is listed in the pricing sheets (please refer to Section 8.2 for details) for each system reported. The prices for all products and features are provided by IBM and available the same day as product or feature availability. All products are currently orderable for delivery on or before the published availability date.

8.2 Five Year Cost of System Configuration

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.

The price sheet for the AS/400e server s40-2208 and associated client systems is contained on the following pages. The discounts used are disclosed below.

Revenue Allowance

This allowance of 29.5% was based on total hardware and software purchase price of the configuration.

Midrange Service Option (MRSO)

This discount is available for customers when agreement is reached for the customer to perform specified service duties (consult marketing representative for details). For the priced configuration, the MSAD discount is 17%.

Extended Maintenance Option (EMO)

This is a discount for prepayment of maintenance costs. A discount of 17% is available for this configuration based on payment for five years maintenance at time of purchase. This discount is applied to the balance after the Midrange Service Option is applied.

8.3 Statement of tpmC and Price/Performance

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

The IBM AS/400e server s40-2208 was measured at 43,169.85 tpmC with a 5-year system price of $5,564,844. The respective price-performance for the AS/400e server s40-2208 is $128 per tpmC. The AS/400e server s40-2208 priced configuration is currently orderable for delivery on or after September 11, 1998.

8.3.1 IBM AS/400e server s40-2208 Five-Year System Price Configuration

AS/400e server model s40-2208	TPC-C REV 3.4 EXECUTIVE SUMMARY
	Report Date: September 1, 1997

Token-Ring Adapter (9249)	Included in S40 base price		1	$0		$0
2048 Optional MB Base Memory	8193	$39,424	4	$157,696		$157,696
Optical Link	2688	$2,000	6	$12,000		$12,000
Optical Bus Adapter	2695	$1,000	1	$1,000		$1,000
2048 MB Main Storage	3193	$45,056	16	$720,896		$720,896
Storage Expansion Unit	5057	$5,000	1	$5,000	$10,560	$15,560
Storage Expansion Unit	5058	$5,000	11	$55,000	$116,160	$171,160
1063 Mbps System Unit Expansion Tower	5073	$14,900	1	$14,900		$14,900
1063 Mbps Storage Expansion Tower	5083	$14,900	10	$149,000	$144,000	$293,000
Token Ring Adapter	6149	$1,200	7	$8,400		$8,400
RAID Disk Unit Controller	6533	$9,900	21	$207,900		$207,900
4 GB 1/4-inch Tape Drive	6382	$1,800	1	$1,800		$1,800
RPQ 843804	843804	$900	1	$900		$900
LAN/WAN/Workstation IOP	2629	$2,600	7	$18,200		$18,200
V.24./EIA232 20 ft Cable	0330	$125	1	$125		$125
17.2 GB Disk Unit	6714	$3,800	40	$152,000		$152,000
8.58 GB Disk Unit	6713	$2,100	315	$661,500		$661,500
Base 8.58 GB Disk Unit	8713	$800	1	$800		800
Server Subtotal				*$2,647,117*	*$362,592*	*$3,009,709*
Client Hardware:						
AS/400 9401-150 Growth Server Pkg V4R1	0194	$12,000	1	$12,000	$4,272	$16,272
4.194 GB Disk Unit	Included in S40 base price		1	$0		$0
128 MB Base Memory	Included in S40 base price		1	$0		$0
Twinax Work Station Controller	Included in S40 base price		1	$0		$0
Base Token Ring Adapter (9724)	Included in S40 base price		1	$0		$0
Tape Drive	Included in S40 base price		1	$0		$0
64 MB Additional Main Storage Memory	3110	$768	1	$768		$768
Token Ring Adapter	2724	$840	1	$840		$840
4.194 GB Single Disk Unit	6607	$1,300	1	$1,300		$1,300
Single Client Subtotal				*$14,908*	*$4,272*	*$19,180*
Number of Clients			97			
Client Subtotal				*$1,446,076*	*$414,384*	*$1,860,460*
Server Software:						
IBM Operating System/400 V4R1M0 60-user	Included in S		1	$0		$0
Client Software:						
IBM Operating System/400 V4R1M0	Included in S		97	$0		$0
CICS for OS/400	5716-DFH	$2,450	97	$237,650		$237,650
ILE COBOL	5738-CB1	$900	1	$900		$900
Application Development Toolkit	5716-PW1	$900	1	$900		$900
DB2 Query Manager Dev. Toolkit	Included in S		1	$0		$0
ILE C	5716-CX2	$900	1	$900		$900
Software Subtotal				*$240,350*		*$240,350*
User Connectivity:						
16-port Token Ring Switch (including spare)	8272-216	$9,995	7	$69,967		$69,967
4-port UFC for 8272 (10% spares)	9196	$1,600	12	$19,200		$19,200
TOKEN RING CAU (including spare)	8230-013	$3,340	5	$16,698		$16,698
4-port LIU for 8230 (including spare)	2009	$300	17	$5,100		$5,100
20-PORT TOKEN RING LAM for 8030 (including spar	6738	$1,830	6	$10,981		$10,981
8-port Token Ring connection unit (10% spares)	8226	$545	4915	$2,678,675		$2,678,675
User Connectivity Subtotal				*$2,800,621*		*$2,800,621*
5-year System Subtotal				*$7,134,164*	*$776,976*	*$7,911,140*
Discounts						
Revenue Allowance				($2,104,578)		*($2,104,578)*
Midrange System Option (MSRO)					($132,086)	*($132,086)*
Extended Maintenance Option (EMO)					($109,631)	*($109,631)*
5-year System Total				$5,029,586	$535,259	**$5,564,844**

Notes:

Revenue Allowance is applied to hardware and software for the priced configuration.
EMO is a discount for prepayment of 5 years of maintenace costs.
MSRO is available to customers when agreement is reached for the customer to perform certain duties

Five-Year Cost of Ownership: $5,564,844

tpmC Rating: 43,169.85

$/tpmC: $128.91

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

9.0 Clause 8: Audit - Related Items

If the benchmark has been independently audited, then the auditor's name, address, phone number, and a brief audit summary report indicating compliance must be included in the Full Disclosure Report. A statement should be included, specifying when the complete audit report will become available and who to contact in order to obtain a copy.

These TPC Benchmark™ C results have been audited by Francis Raab of Information Paradigm. The attestation letter is included at the end of this report.

Appendix A. System Parameters and User Profile

A.1 System Parameters

Table 4 shows the system parameters changed for these TPC-C measurements.

Table 4. System Parameters	
System Parameters	**AS/400e server s40-2208**
QMCHPOOL	3,163,428
*BASE	38,779,612
PAGING OPTION	USRDFN
QBASACTLVL	32,000
QPFRADJ	0
EDTRCYAP	*OFF

A.1.1 Transaction Subsystem Description

All transactions and associated user processes ran in the *BASE memory pool.

A.2 User Profile

All users executing the TPC Benchmark™ C transactions signed on using the same User Profile. The User Profile defines the libraries included in the library list, defines which library is the current library, special authorities granted to this user, and which program is called when the user signs on (FIRSTPGM1).

Appendix B: Database File Definitions

B.1 AREFFIL: Reference File

```
A*   OCO Source Materials
A* The Source code for this program is not published or otherwise
A* divested of its trade secrets, irrespective of what has been
A* deposited with the U.S. Copyright office
A* (C) Copyright IBM Corp. 1993, 1996, 1998

A*B:H2.AREFFIL:  Reference File
          R REFRCD                    TEXT('TPCC Reference File')
A*This file contains the field definition for most of the other
A*files used in the TPC-C benchmark.  For example to define the
A*warehouse ID field, both the warehouse file (WRHS) and the
A*district file (DSTRCT) reference WRHSID in this file.  This
A*helps when field definitions need to change, the change only
A*needs to be made in one place.
A*
A*UPDATED:10/23/93
A*Changes BALANC and CRDLMT to 12 digit fields as required
A*by the TPC-C V1.1 Specification
A*UPDATED:03/04/96
A*Changes BALANC and CRDLMT to 13 digit fields for effieiency.
A*Added ITMIMAGE perf v3.0 spec.  Converted many fields to binary.
A* wrhsid, distid, custid, itemid, ordid, oline, ytd2, qty2, qty4
A*
A*UPDATED:03/31/98
A*
A*
A*Code Identifiers
A* Note Binary nubers are 4B for two byte shorts and 9B
A* for long integer 32 byte numbers comforms to SQL defintions
A*
A            WRHSID         4B         TEXT('Warehouse ID')
A                                      COLHDG('W/H ID')
A            DISTID         4B         TEXT('District ID')
A                                      COLHDG('District')
A                                      EDTCDE(1)
A            CUSTID         9B         TEXT('Customer ID')
A                                      COLHDG('Customer')
A            ITEMID         9B         TEXT('Item ID')
A                                      COLHDG('Item')
A            ITMIMAGE       9B         TEXT('ImageID')
A                                      COLHDG('Image#')
A            ORDID          9B         TEXT('Order ID')
A                                      COLHDG('Order #')
A                                      EDTCDE(4)
A*                                     EDTWRD('       0')
A            OLINE          3P         TEXT('Order Line Number')
A                                      COLHDG('Order' 'Line #')
A*Name, Address and Descriptor Information
A            LOCNAM        10          TEXT('Name of W/H or District')
A                                      COLHDG('Location' 'Name')
A            FNAME         16          TEXT('First Name')
A                                      COLHDG('First' 'Name')
A            MINIT          2          TEXT('Middle Initial')
A                                      COLHDG('Middle' 'Init')
A            LNAME         16          TEXT('Last Name')
A                                      COLHDG('Last' 'Name')
A            ADDR1         20          TEXT('Address Line 1')
A                                      COLHDG('Address' 'Line 1')
A            ADDR2         20          TEXT('Address Line 2')
A                                      COLHDG('Address' 'Line 2')
A            CITY          20          TEXT('City')
A            STATE          2          TEXT('State')
A            ZIPCD          9A         TEXT('Zip Code')
A                                      COLHDG('Zip' 'Code')
A            PHONE         16          TEXT('Phone Number')
A            ITMNAM        24          TEXT('Item Name')
A                                      COLHDG('Item' 'Name')
A            CARRID         2          TEXT('Carrier ID')
A                                      COLHDG('Carrier' 'Number')
A            LOCAL          1  0       TEXT('Flag to indicate local')
A*Financial and Inventory Numbers
A            TAX            5  4       TEXT('Tax Percentage')
A                                      EDTWRD('0.    ')
A            DSCNT          5  4       TEXT('Discount Percentage')
A                                      COLHDG('Discount')
A                                      EDTWRD('0.    ')
A*UPDATED:03/01/96


  A            BALANC        13  2      TEXT('Balance Information')
A                                      COLHDG('Balance')
A                                      EDTWRD('        $0  .  -')
A            CREDIT         2          TEXT('Credit Status-GC or BC')
A                                      COLHDG('Credit' 'Status')
A*UPDATED:03/01/96
A            CRDLMT        13  2       TEXT('Credit Limit')
A                                      COLHDG('Credit' 'Limit')
A                                      EDTWRD('        $0  .  -')
A            AMOUNT7        7P 2       TEXT('Amount')
A                                      EDTWRD('  $0  .  -'  )
A            YTD           13  2       TEXT('YTD Amount')
A                                      EDTWRD('        $0  .  -'  )
A            YTD2           9B         TEXT('YTD Amount')
A                                      EDTCDE(4)
A            QTY2           3P 0       TEXT('Quantity')
A                                      COLHDG('Quantity')
A                                      EDTCDE(1)
A            QTY4           3P 0       TEXT('Quantity')
A                                      COLHDG('Quantity')
A                                      EDTCDE(1)
A            QTY3           3  0       TEXT('Quantity')
A                                      COLHDG('Quantity')
A                                      EDTCDE(1)
A            PRICE          5  2       TEXT('Price')
A                                      EDTWRD('$0  .  -')
A*Date and Time Information
A            TIMEDATE       8          TEXT('Internal Time Date')
A            DATE           8S         TEXT('Date')
A                                      EDTWRD('  -  -    ')
A            TIME           6S         TEXT('Time')
A                                      EDTWRD('  :  :  ')
A*Pad Information
A            CDATA        500          TEXT('Customer Data')
A                                      COLHDG('Cust' 'Filler')

A            HDATA         24          TEXT('History Information')
A                                      COLHDG('Hist' 'Filler')

A            IDATA         50          TEXT('Item Data')
A                                      COLHDG('Item' 'Filler')
A            DISTINFO      24          TEXT('Dist Info')
A                                      COLHDG('Dist' 'Info')
A*TPCC - Plus Information
A            CPHNTC        10          COLHDG('Phonetic Search')
A            DSPDID         2  0       TEXT('District ID')
A                                      COLHDG('District')
A                                      EDTCDE(4)
A            DSPOID         8  0       TEXT('Order ID')
A                                      COLHDG('Order #')
A                                      EDTCDE(4)
A            DSPOLN         2  0       TEXT('Order Line Number')
A                                      COLHDG('Order' 'Line #')
A            AMOUNT6        6  2       TEXT('Amount')
A                                      EDTWRD('$0   .  -')
A            DSPQTY         3S 0       TEXT('Quantity')
A                                      COLHDG('Quantity')
A                                      EDTCDE(1)
```

B.2 CSTMRLFCRT: Customer Logical File

```
A          R CSRCD                     PFILE(CSTMRPF)
A          K CWID
A          K CDID
A          K CLAST
A          K CFIRST
```

B.3 CSTMRLFNAM: Customer Names Logical File

```
A          R CSRCD                     PFILE(CSTMRPF)
A          K CWID
A          K CDID
A          K CLAST
```

B.4 CSTMR: Customer Logical File

```
A*This is the file definition for the customer data base file.
A*All fields that are stored in the customer file are defined
A*here, the actual field definitions are in AREFFIL (the
A*reference file) and are pulled into this file when it is
A*compiled (CRTPF command executed against this source).
A*The UNIQUE keyword guarantees that there will be not duplicate
A*key values inserted into the file.
A*
A                                      UNIQUE
A          R CSRCD                     PFILE(CSTMRPF)
A                                      TEXT('CUSTOMER MASTER FILE - TPCC')
A          K CID
A          K CDID
A          K CWID
```

B.5 CSTMRPF: Customer Physical File

```
  A                                    REF(AREFFIL)
A*This is the file definition for the customer data base file.
A*All fields that are stored in the customer file are defined
A*here, the actual field definitions are in AREFFIL (the
A*reference file) and are pulled into this file when it is
A*compiled (CRTPF command executed against this source).
A          R CSRCD                     TEXT('CUSTOMER MASTER FILE - TPCC')
A            CID           R           REFFLD(CUSTID)
A            CDID          R           REFFLD(DISTID)
A            CWID          R           REFFLD(WRHSID)
A            CFIRST        R           REFFLD(FNAME)
A            CINIT         R           REFFLD(MINIT)
A            CLAST         R           REFFLD(LNAME)
A            CLTOD         R           REFFLD(TIMEDATE)
A                                      COLHDG('Date of' 'Last Order')
A                                      TEXT('Date of DB Build')
A            CADDR1        R           REFFLD(ADDR1)
A            CCREDT        R           REFFLD(CREDIT)
A                                      COLHDG('Credit' 'Status')
A                                      TEXT('Credit Status')
A            CADDR2        R           REFFLD(ADDR2)
A            CDCT          R           REFFLD(DSCNT)
A            CCITY         R           REFFLD(CITY)
A            CSTATE        R           REFFLD(STATE)
A            CZIP          R           REFFLD(ZIPCD)
A            CPHONE        R           REFFLD(PHONE)
A            CBAL          R           REFFLD(BALANC)
A                                      TEXT('Customer Balance')
A            CCRDLM        R           REFFLD(CRDLMT)
A                                      TEXT('Credit Limit')
A            CYTD          R           REFFLD(YTD)
A                                      TEXT('Customer YTD')
A            CPAYCNT       R           REFFLD(QTY4)
```

```
A                                       TEXT('Customer Payments')
A            CDELCNT   R                REFFLD(QTY4)
A                                       TEXT('Customer Deliveries')
A*           CLTIME    R                REFFLD(TIME)
A*                                      COLHDG('Time of' 'Last Order')
A*                                      TEXT('Time of DB Build')
A            CDATA     R
```

B.6 DSTRCT: District File

```
                                        REF(AREFFIL)
     A*This is the file definition for the district data base file.
     A*All fields that are stored in the district file are defined
     A*here, the actual field definitions are in AREFFIL (the
     A*reference file) and are pulled into this file when it is
     A*compiled (CRTPF command executed against this source).
     A*The UNIQUE keyword guarantees that there will be not duplicate
     A*key values inserted into the file.
     A*
     A                                  UNIQUE
     A          R DSRCD                 TEXT('District Master File - TPCC')
     A            DID       R          REFFLD(DISTID)
     A            DWID      R          REFFLD(WRHSID)
     A            DNAME     R          REFFLD(LOCNAM)
     A                                  COLHDG('District' 'Name')
     A            DADDR1    R          REFFLD(ADDR1)
     A            DADDR2    R          REFFLD(ADDR2)
     A            DCITY     R          REFFLD(CITY)
     A            DSTATE    R          REFFLD(STATE)
     A            DZIP      R          REFFLD(ZIPCD)
     A            DTAX      R          REFFLD(TAX)
     A            DYTD      R          REFFLD(YTD)
     A                                  COLHDG('YTD' 'Balance')
     A                                  TEXT('YTD Balance')
     A            DNXTOR    R          REFFLD(ORDID)
     A                                  COLHDG('Next' 'Order #')
     A                                  TEXT('Next Order Number')
     A          K DID
     A          K DWID
```

HSTRY: History File

```
                                        REF(AREFFIL)
     A          R HSRCD                 TEXT('History File for TPCC')
     A*This is the file definition for the history data base file.
     A*All fields that are stored in the history file are defined
     A*here, the actual field definitions are in AREFFIL (the
     A*reference file) and are pulled into this file when it is
     A*compiled (CRTPF command executed against this source).
     A*The UNIQUE keyword is not needed in this file because there
     A*are no key fields for this file.
     A*
     A            HDID      R          REFFLD(DISTID)
     A            HWID      R          REFFLD(WRHSID)
     A            HCID      R          REFFLD(CUSTID)
     A            HCDID     R          REFFLD(DISTID)
     A            HCWID     R          REFFLD(WRHSID)
     A            HTOD      R          REFFLD(TIMEDATE)
     A                                  COLHDG('Payment' 'Time Date')
     A*           HTIME     R          REFFLD(TIME)
     A*                                 COLHDG('Payment' 'Time')
     A            HAMT      R          REFFLD(AMOUNT7)
     A                                  COLHDG('Payment' 'Amount')
     A                                  TEXT('Payment Amount')
     A            HDATA     R          REFFLD(HDATA)
```

ITEM: Item Physical File

```
     A                                  REF(AREFFIL)
     A*This is the file definition for the item data base file.
     A*All fields that are stored in the item file are defined
     A*here, the actual field definitions are in AREFFIL (the
     A*reference file) and are pulled into this file when it is
     A*compiled (CRTPF command executed against this source).
     A*The UNIQUE keyword guarantees that there will be not duplicate
     A*key values inserted into the file.
     A*
     A                                  UNIQUE
     A          R ITRCD                 TEXT('Item File for TPCC')
     A            IID       R          REFFLD(ITEMID)
     A            IMAGEID   R          REFFLD(ITMIMAGE)
     A            INAME     R          REFFLD(ITMNAM)
     A            IPRICE    R          REFFLD(PRICE)
     A            IDATA     R
     A          K IID
```

B.7 ITEMLF: Item Logical File

```
     A          R ITRCD   PFILE(ITEM)
     A          K IID
```

B.8 NEWORD: New Order Logical File

```
     A*This is the file definition for the new order data base file.
     A*All fields that are stored in the new order file are defined
     A*here, the actual field definitions are in AREFFIL (the
     A*reference file) and are pulled into this file when it is
     A*compiled (CRTPF command executed against this source).
     A*The UNIQUE keyword guarantees that there will not be duplicate
     A*key values inserted into the file.
     A*
     A                                  UNIQUE
     A          R NORCD                 PFILE(NEWORDPF)
                                        TEXT('New Orders File - TPCC')
     A          K NOWID
     A          K NODID
     A          K NOOID
```

B.9 NEWORDLF: New Order Logical File

```
A          R NORCD                     PFILE(NEWORDPF)
A          K NOWID
A          K NODID
```

B.10 NEWORDPF: New Order Physical File

```
                                       REF(AREFFIL)
A*This is the file definition for the new order data base file.
A*All fields that are stored in the new order file are defined
A*here, the actual field definitions are in AREFFIL (the
A*reference file) and are pulled into this file when it is
A*compiled (CRTPF command executed against this source).
A          R NORCD                     TEXT('New Orders File - TPCC')
A            NOOID     R               REFFLD(ORDID)
A            NODID     R               REFFLD(DISTID)
A            NOWID     R               REFFLD(WRHSID)
```

B.11 ORDERS: Orders Logical File

```
A*B:H2.ORDERS:  Orders File
A*This is the file definition for the orders data base file.
A*All fields that are stored in the orders file are defined
A*here, the actual field definitions are in AREFFIL (the
A*reference file) and are pulled into this file when it is
A*compiled (CRTPF command executed against this source).
A          R ORRCD                     PFILE(ORDERSPF)
A            OWID
A            ODID
A            OCID                      EDTWRD('0        ')
A            OID
A            OENTTOD
A*           OENTTM
A            OCARID
A            OLINES
A            OLOCAL
A          K OWID
A          K ODID
A          K OCID
A          K OID
```

B.12 ORDERSLF: Orders Logical File

```
A                                      UNIQUE
A          R ORRCD                     PFILE(ORDERSPF)
A          K OWID
A          K ODID
A          K OID
```

B.13 ORDERSPF: Orders Physical File

```
                                       REF(AREFFIL)
A*This is the file definition for the orders data base file.
A*All fields that are stored in the orders file are defined
A*here, the actual field definitions are in AREFFIL (the
A*reference file) and are pulled into this file when it is
A*compiled (CRTPF command executed against this source).
A          R ORRCD                     TEXT('Orders Master File for TPCC')
A            OWID      R               REFFLD(WRHSID)
A            ODID      R               REFFLD(DISTID)
A            OCID      R               REFFLD(CUSTID)
A            OID       R               REFFLD(ORDID)
A            OENTTOD   R               REFFLD(TIMEDATE)
A                                      COLHDG('Order' 'Date')
A                                      TEXT('Order Entry Time Date')
A*           OENTTM    R               REFFLD(TIME)
A*                                     COLHDG('Order' 'Time')
A*                                     TEXT('Order Entry Time')
A            OCARID    R               REFFLD(CARRID)
A            OLINES    R               REFFLD(OLINE)
A                                      COLHDG('Number of' 'Order Lines')
A                                      TEXT('Number of Lines in Order')
A            OLOCAL    R               REFFLD(LOCAL)
A                                      TEXT('Flag to indicate local order')
```

B.14 ORDLIN: Order Lines Logical File

```
A*This is the file definition for the orders data base file.
A*All fields that are stored in the orders file are defined
A*here, the actual field definitions are in AREFFIL (the
A*reference file) and are pulled into this file when it is
```

```
A*compiled (CRTPF command executed against this source).
A          R ORRCD                     PFILE(ORDERSPF)
A            OWID
A            ODID
A            OCID                      EDTWRD('0        ')
A            OID
A            OENTTOD
A*           OENTTM
A            OCARID
A            OLINES
A            OLOCAL
A          K OWID
A          K ODID
A          K OCID
A          K OID
```

B.15 ORDLINLF: Order Lines Logical File

```
A          R OLRCD                     PFILE(ORDLINPF)
A          K OLWID
A          K OLDID
A          K OLOID
```

B.16 ORDERLINPF: Order Lines Physical File

```
A                                        REF(AREFFIL)
A          R OLRCD                     TEXT('Order Line File for TPCC')
A            OLOID     R               REFFLD(ORDID)
A            OLDID     R               REFFLD(DISTID)
A                                      TEXT('Customers District')
A            OLWID     R               REFFLD(WRHSID)
A                                      COLHDG('Customers' 'Warehouse')
A                                      TEXT('Customers Warehouse')
A            OLNBR     R               REFFLD(OLINE)
A                                      TEXT('Line Number of Order')
A            OLSPWH    R               REFFLD(WRHSID)
A                                      COLHDG('Supply' 'Warehouse')
A                                      TEXT('Supply Warehouse ID')
A            OLIID     R               REFFLD(ITEMID)
A                                      TEXT('Item Ordered')
A            OLQTY     R               REFFLD(QTY2)
A                                      COLHDG('Qty' 'Ordered')
A                                      TEXT('Quantity Ordered')
A            OLAMNT    R               REFFLD(AMOUNT7)
A                                      TEXT('Order Line Amount')
A            OLDLVTOD  R               REFFLD(TIMEDATE)
A                                      COLHDG('Delivery' 'Time Date')
A                                      TEXT('Delivery Time Date')
A*           OLDLVT    R               REFFLD(TIME)
A*                                     COLHDG('Delivery' 'Time')
A*                                     TEXT('Delivery Time')
A            OLDSTI    R               REFFLD(DISTINFO)
A                                      TEXT('Dist Information')
```

B.17 STOCK: Stock Logical File

```
A                                      UNIQUE
A          R STRCD                     PFILE(STOCKPF)
A          K STWID
A          K STIID
```

B.18 STOCKLF: Stock Logical File

```
A          R STRCD                     TEXT('Stock File for TPCC')
A                                      PFILE(STOCKPF)
A          K STWID
A          K STIID
```

B.19 STOCKPF: Stock Physical File

```
A*B:H2.STOCKPF:  Stock File
A                                      REF(AREFFIL)
A          R STRCD                     TEXT('Stock File for TPCC')
A            STWID     R               REFFLD(WRHSID)
A            STIID     R               REFFLD(ITEMID)
A                                      TEXT('Warhouse Nbr of Stock Item')
A            STQTY     R               REFFLD(QTY4)
A                                      COLHDG('Qty in' 'Stock')
A                                      TEXT('Quantity on hand')
A            STDI01    R               REFFLD(DISTINFO)
A                                      TEXT('Dist Info 1')
A            STDI02    R               REFFLD(DISTINFO)
A                                      TEXT('Dist Info 2')
A            STDI03    R               REFFLD(DISTINFO)
A                                      TEXT('Dist Info 3')
A            STDI04    R               REFFLD(DISTINFO)
A                                      TEXT('Dist Info 4')
A            STDI05    R               REFFLD(DISTINFO)
A                                      TEXT('Dist Info 5')
A            STDI06    R               REFFLD(DISTINFO)
A                                      TEXT('Dist Info 6')
A            STDI07    R               REFFLD(DISTINFO)
A                                      TEXT('Dist Info 7')
A            STDI08    R               REFFLD(DISTINFO)
A                                      TEXT('Dist Info 8')
A            STDI09    R               REFFLD(DISTINFO)
A                                      TEXT('Dist Info 9')
A            STDI10    R               REFFLD(DISTINFO)
A                                      TEXT('Dist Info 10')
A            STYTD     R               REFFLD(YTD2)
A                                      TEXT('Qty ordered YTD')
A            STORDRS   R               REFFLD(QTY4)
A                                      TEXT('# times ordered')
A            STREMORD  R               REFFLD(QTY4)
A                                      TEXT('# times ordered remotely')
A            STDATA    R               REFFLD(IDATA)
```

B.20 WRHS: Warehouse Physical File

```
                                       REF(AREFFIL)
A                                      UNIQUE
A          R WRRCD                     TEXT('Warehouse Master File - TPCC')
A            WID       R               REFFLD(WRHSID)
A            WNAME     R               REFFLD(LOCNAM)
A                                      COLHDG('W/H' 'Name')
A            WADDR1    R               REFFLD(ADDR1)
A            WADDR2    R               REFFLD(ADDR2)
A            WCITY     R               REFFLD(CITY)
A            WSTATE    R               REFFLD(STATE)
A            WZIP      R               REFFLD(ZIPCD)
A            WTAX      R               REFFLD(TAX)
A            WYTD      R               REFFLD(YTD)
A                                      COLHDG('YTD' 'Balance')
A                                      TEXT('Warehouse YTD Balance')
A          K WID
```

Appendix C. Database Build Programs

Program Flow For Build of Server, Client, and Database

This list shows the order of invocation of the database build programs. Each level of indentation is a call. If a program is indented more than the previous program, it was called by the previous program. If it is at the same margin as the previous program (or as earlier program), it was called by the same program as the previous (or earlier) program. To make determining the level of call easier, the level number will follow the program enclosed in parenthesis.

BLDTPCCPGM (2)
- CRTBLDPGMS (3)
  - NATBLD (4)
  - CRTHASH (4)
    - C_CREATE (5)
    - C_CREATE2 (5)
    - C_CREATE3 (5)
- DLTLOGICLS (3)
- LOADTPCCF (3)
  - LOADW_D (4)
  - LOADITEM (4)
  - LOADCST (4)
  - LOADORD (4)
- SETUP (3)
  - ORDERSVIEW (4)
  - ORDLINVIEW (4)
  - NEWORDVIEW (4)
  - CSTMRVIEW (4)
  - STRJRNTPCC (4)
- CRTENVPGMS (3)

BLDTPCCPGM: Database Build Programs

```
            PGM         PARM(&DB &ENV &CRTSAVFILE &DB2)
           /* This program brings in the number of warehouses and  */
           /* the options to compile the build programs and the    */
           /* environment programs.  The data library will get     */
           /* then get built, if it already exists, it is deleted  */
           /* and rebuilt.                                         */

           DCL         VAR(&WRHS) TYPE(*DEC) LEN(4 0)
           DCL         VAR(&DB) TYPE(*CHAR) LEN(1)
           DCL         VAR(&DB2) TYPE(*CHAR) LEN(1)
           DCL         VAR(&ENV) TYPE(*CHAR) LEN(1)
           DCL         VAR(&APPLIB) TYPE(*CHAR) LEN(10)
           DCL         VAR(&DATLIB) TYPE(*CHAR) LEN(10)
           DCL         VAR(&DB2SETPGM) TYPE(*CHAR) LEN(10)
           DCL         VAR(&LIBRARY) TYPE(*CHAR) LEN(10)
           DCL         VAR(&LIBASP) TYPE(*CHAR) LEN(1)
           DCL         VAR(&BLDLIB) TYPE(*CHAR) LEN(10)
           DCL         VAR(&CRTSAVFILE) TYPE(*CHAR) LEN(1)
           DCL         VAR(&SAVLIB) TYPE(*CHAR) LEN(10)
           DCL         VAR(&JRNLIB) TYPE(*CHAR) LEN(10)
           DCL         VAR(&WRHSCHAR) TYPE(*CHAR) LEN(4)
           DCL         VAR(&STRWHCHAR) TYPE(*CHAR) LEN(4)
           DCL         VAR(&ENDWHCHAR) TYPE(*CHAR) LEN(4)
           DCL         VAR(&ENDWRHS) TYPE(*CHAR) LEN(4)
           DCL         VAR(&STRWRHS) TYPE(*CHAR) LEN(4)
           DCL         VAR(&STRWH) TYPE(*DEC) LEN(4 0) VALUE(1)
           DCL         VAR(&ENDWH) TYPE(*DEC) LEN(4 0)
           DCL         VAR(&JRNASP) TYPE(*CHAR) LEN(1)
           DCL         VAR(&DATE) TYPE(*CHAR) LEN(6)
           DCL         VAR(&PRVLIBVAL) TYPE(*CHAR) LEN(10) /* +
                         Previous TPC-C Library Names */

           MONMSG      MSGID(CPF2556)

SETRPY:    ADDRPYLE    SEQNBR(111) MSGID(CPA7025) RPY(I)
           MONMSG      MSGID(CPF2555) EXEC(DO)
           RMVRPYLE    SEQNBR(111)
           RMVRPYLE    SEQNBR(112)
           RMVRPYLE    SEQNBR(113)
           RMVRPYLE    SEQNBR(114)
           RMVRPYLE    SEQNBR(115)
           ADDRPYLE    SEQNBR(111) MSGID(CPA7025) RPY(I)
           ENDDO
           ADDRPYLE    SEQNBR(112) MSGID(CPA1332) RPY(I)
           ADDRPYLE    SEQNBR(113) MSGID(CPF2110) RPY(I)
           ADDRPYLE    SEQNBR(114) MSGID(CPF1054) RPY(I)
           ADDRPYLE    SEQNBR(115) MSGID(CPF2105) RPY(I)
           CHGJOB      JOB(*) INQMSGRPY(*SYSRPYL)


GETPARMS:  RTVDTAARA   DTAARA(TPCCINFO/DATALIB) RTNVAR(&DATLIB)
           RTVDTAARA   DTAARA(TPCCINFO/LIBASP) RTNVAR(&LIBASP)
           RTVDTAARA   DTAARA(TPCCINFO/APPLIB) RTNVAR(&APPLIB)
           RTVDTAARA   DTAARA(TPCCINFO/MNTLIB) RTNVAR(&BLDLIB)
           RTVDTAARA   DTAARA(TPCCINFO/SAVELIB) RTNVAR(&SAVLIB)
           RTVDTAARA   DTAARA(TPCCINFO/JRNLIB) RTNVAR(&JRNLIB)
           RTVDTAARA   DTAARA(TPCCINFO/JRNASP) RTNVAR(&JRNASP)
           RTVDTAARA   DTAARA(TPCCINFO/ENDWRHS) RTNVAR(&ENDWRHS)
           RTVDTAARA   DTAARA(TPCCINFO/STRWRHS) RTNVAR(&STRWRHS)
           CHGVAR      VAR(&WRHS) VALUE(&ENDWRHS)
           CHGVAR      VAR(&STRWH) VALUE(&STRWRHS)
           RTVSYSVAL   SYSVAL(QDATE) RTNVAR(&DATE)

           CHGVAR      VAR(&WRHSCHAR) VALUE(&WRHS)
           CHGVAR      VAR(&ENDWH) VALUE(&WRHS)
           CHGVAR      VAR(&STRWHCHAR) VALUE(&STRWH)
           CHGVAR      VAR(&ENDWHCHAR) VALUE(&ENDWH)

           MONMSG      MSGID(CPF0000)
           ENDSBS      SBS(TPCCBLDSBS) OPTION(*IMMED)
           MONMSG      MSGID(CPF1054)
           DLTSBSD     SBSD(&BLDLIB/TPCCBLDSBS)
           MONMSG      MSGID(CPF2105)
           DLTJOBD     JOBD(&BLDLIB/TPCBLDJOBD)
           MONMSG      MSGID(CPF2105)
           CRTSBSD     SBSD(&BLDLIB/TPCCBLDSBS) POOLS((1 *BASE)) +
                         TEXT('TPCCBLD Jobs Subsystem')
           DLTCLS      CLS(&BLDLIB/TPCCBLDCLS)
           MONMSG      MSGID(CPF2105)
           CRTCLS      CLS(&BLDLIB/TPCCBLDCLS) RUNPTY(21) +
                         TIMESLICE(2000) PURGE(*NO)
           DLTJOBQ     JOBQ(&BLDLIB/TPCBLDJOBQ)
           MONMSG      MSGID(CPF2105)
           CRTJOBQ     JOBQ(&BLDLIB/TPCBLDJOBQ)
           ADDJOBQE    SBSD(&BLDLIB/TPCCBLDSBS) +
                         JOBQ(&BLDLIB/TPCBLDJOBQ) MAXACT(*NOMAX)
           CRTJOBD     JOBD(&BLDLIB/TPCBLDJOBD) +
                         JOBQ(&BLDLIB/TPCBLDJOBQ) RTGDTA(TPCCBLD) +
                         INLLIBL(&BLDLIB QTEMP QGPL) LOG(4 +
                         0 *SECLVL) INQMSGRPY(*DFT)
           ADDRTGE     SBSD(&BLDLIB/TPCCBLDSBS) SEQNBR(99) +
                         CMPVAL(*ANY) PGM(QCMD) +
                         CLS(&BLDLIB/TPCCBLDCLS)
           STRSBS      SBSD(&BLDLIB/TPCCBLDSBS)
CHKDBLIB:  IF (&DB *EQ 'Y') THEN(DO)
           CHKOBJ      OBJ(&DATLIB) OBJTYPE(*LIB)
           MONMSG      MSGID(CPF9801) EXEC(CRTLIB LIB(&DATLIB) +
                         ASP(&LIBASP))
           ENDJRNPF    FILE(*ALL) JRN(&DATLIB/TPCCJRN)
           MONMSG      MSGID(CPF0000)
           IF (&STRWRHS *EQ '1') THEN(DO)
           CLRLIB      LIB(&DATLIB)
           CHGLIB      LIB(&DATLIB) TEXT('Data library ' *CAT &DATE)
           ENDDO
           IF (&STRWRHS *NE '1') THEN(GOTO EXTENDDB)
           ENDDO
CHKOBJLIB: ADDLIBLE    LIB(&APPLIB)
           MONMSG      MSGID(CPF2103)
           MONMSG      MSGID(CPF2110) EXEC(DO)
           CRTLIB      LIB(&APPLIB) TEXT('Application Library ' +
                         *CAT &DATE)
             ADDLIBLE  LIB(&APPLIB)
           ENDDO

           SNDPGMMSG   MSGID(CPF9898) MSGF(QCPFMSG) +
                         MSGDTA('Starting Build') +
                         TOPGMQ(*EXT) MSGTYPE(*STATUS)


           /* Check for DB Build */

DBCHECK:   IF          COND(&DB *NE 'Y') THEN(GOTO CMDLBL(SKIPDB))
           CRTCLPGM    PGM(&BLDLIB/CRTBLDPGMS) +
                         SRCFILE(&BLDLIB/QCLSRC) +
                         SRCMBR(CRTBLDPGMS) OPTION(*NOSOURCE)
           CALL        PGM(&BLDLIB/CRTBLDPGMS) PARM(&STRWH &ENDWH +
                         &BLDLIB &DATLIB)

           CALL        PGM(&BLDLIB/CRTHASHES)

           GOTO        CMDLBL(SBMTPCCF)

EXTENDDB:  ADDLIBLE    LIB(&BLDLIB)
           MONMSG      MSGID(CPF2103)

/* DELETE THE LOGICALS TO SPEED UP THE BUILD                        */

           CRTCLPGM    PGM(&BLDLIB/DLTLOGICLS) +
                         SRCFILE(&BLDLIB/QCLSRC) OPTION(*NOSOURCE)

           CALL        PGM(&BLDLIB/DLTLOGICLS) PARM(&DATLIB)

/*                 FILL DATA FILES                                  */
 SBMTPCCF:  CALL        PGM(&BLDLIB/LOADTPCCF) PARM(&DATLIB &BLDLIB +
                         &STRWHCHAR &WRHSCHAR)

/*   INITIALIZE APPLICATION ENVIRONMENT                             */

           IF          COND(&DB2 *EQ 'y') THEN(DO)
           RTVDTAARA   DTAARA(TPCCINFO/DB2SETPGM) RTNVAR(&DB2SETPGM)
           RTVDTAARA   DTAARA(TPCCINFO/LIBRARY) RTNVAR(&LIBRARY)
           SBMJOB      CMD(CALL PGM(&LIBRARY/DB2SETPGM) PARM(&BLDLIB +
                         &BLDLIB &DATLIB &CRTSAVFILE &SAVLIB +
                         &JRNLIB &JRNASP &STRWHCHAR &ENDWHCHAR)) +
                         JOB(SETUP) JOBQ(&BLDLIB/TPCBLDJOBQ)
           ENDDO
           ELSE        CMD(DO)
           SBMJOB      CMD(CALL PGM(&BLDLIB/SETUP) PARM(&BLDLIB +
                         &BLDLIB &DATLIB &CRTSAVFILE &SAVLIB +
                         &JRNLIB &JRNASP &STRWHCHAR &ENDWHCHAR)) +
                         JOB(SETUP) JOBQ(&BLDLIB/TPCBLDJOBQ)
           ENDDO
           /* Check for compiling of Environment programs. */
 SKIPDB:   IF          COND(&ENV = 'Y') THEN(DO)

           CHGLIB      LIB(&APPLIB) TEXT('Applications Created on ' +
                         *CAT (&DATE))
           CRTCLPGM    PGM(&BLDLIB/CRTENVPGMS) +
                         SRCFILE(&BLDLIB/QCLSRC) +
                         SRCMBR(CRTENVPGMS) OPTION(*NOSOURCE)
           MONMSG      MSGID(CPF0000) EXEC(SNDPGMMSG MSG('The +
                         application programs could not be created'))
           SBMJOB      CMD(CALL PGM(&BLDLIB/CRTENVPGMS) +
                         PARM(&APPLIB &DATLIB)) JOB(CRTENVPGMS) +
                         JOBQ(QCTL)
           MONMSG      MSGID(CPF0000) EXEC(SNDPGMMSG MSG('The +
                         application programs could not be created'))
           ENDDO

 ENDPGM:   ENDPGM
```

CRTBLDPGMS: Create Database Build Programs

```
           PGM         PARM(&STRWH &ENDWH &BLDLIB &DATLIB)
           DCL         VAR(&STRWH) TYPE(*DEC) LEN(4 0)
           DCL         VAR(&ENDWH) TYPE(*DEC) LEN(4 0)
           DCL         VAR(&BLDLIB) TYPE(*CHAR) LEN(10)
           DCL         VAR(&DATLIB) TYPE(*CHAR) LEN(10)
           DCL         VAR(&VERSION) TYPE(*CHAR) LEN(1)
           DCL         VAR(&VERSIONNUM) TYPE(*DEC) LEN(1 0)
           DCL         VAR(&MSGID) TYPE(*CHAR) LEN(7)
/* This CL program compiles all of the CL, C and CBL programs   */
/* that are needed to do a complete rebuild of the database.    */
/* They are compiled in the order they are needed. None of the  */
/* programs are called from this program, only compiled.        */

           SNDPGMMSG   MSGID(CPF9898) MSGF(QCPFMSG) +
                         MSGDTA('Compiling Build Programs.') +
                         TOPGMQ(*EXT) MSGTYPE(*STATUS)
           CHKOBJ      OBJ(&BLDLIB/TPCCBLDMSG) OBJTYPE(*MSGQ)
           MONMSG      MSGID(CPF9801) EXEC(CRTMSGQ +
                         MSGQ(&BLDLIB/TPCCBLDMSG))
           CLRMSGQ     MSGQ(TPCCBLDMSG)

           CRTCLPGM    PGM(&BLDLIB/NATBLD) SRCFILE(&BLDLIB/QCLSRC) +
                         SRCMBR(NATBLD)

           CRTCLPGM    PGM(&BLDLIB/SETUP) SRCFILE(&BLDLIB/QCLSRC)

           CRTCLPGM    PGM(&BLDLIB/ITEMVIEW) SRCFILE(&BLDLIB/QCLSRC)

           CRTCLPGM    PGM(&BLDLIB/CSTMRVIEW) SRCFILE(&BLDLIB/QCLSRC)

           CRTCLPGM    PGM(&BLDLIB/NEWORDVIEW) SRCFILE(&BLDLIB/QCLSRC)

           CRTCLPGM    PGM(&BLDLIB/ORDERSVIEW) SRCFILE(&BLDLIB/QCLSRC)

           CRTCLPGM    PGM(&BLDLIB/ORDLINVIEW) SRCFILE(&BLDLIB/QCLSRC)

           CRTCLPGM    PGM(&BLDLIB/STOCKVIEW) SRCFILE(&BLDLIB/QCLSRC)

           CRTCLPGM    PGM(&BLDLIB/LOADTPCCF) SRCFILE(&BLDLIB/QCLSRC)

           /* Build Physical Files */
           /* Has to be called before CREATE of LOAD pgms */
           IF          COND(&STRWH *EQ 1) THEN(CALL +
                         PGM(&BLDLIB/NATBLD) PARM(&ENDWH &BLDLIB +
                         &DATLIB))

           ADDLIBLE &DATLIB
           MONMSG      MSGID(CPF0000)

           CRTBNDC     PGM(&BLDLIB/LOADW_D) SRCFILE(&BLDLIB/QCSRC)
           CRTBNDC     PGM(&BLDLIB/LOADITEM) SRCFILE(&BLDLIB/QCSRC)
           CRTBNDC     PGM(&BLDLIB/LOADITONLY) SRCFILE(&BLDLIB/QCSRC)
           CRTBNDC     PGM(&BLDLIB/LOADSTONLY) SRCFILE(&BLDLIB/QCSRC)
           CRTBNDC     PGM(&BLDLIB/LOADCST) SRCFILE(&BLDLIB/QCSRC)
           CRTBNDC     PGM(&BLDLIB/LOADORD) SRCFILE(&BLDLIB/QCSRC)

           ENDPGM
```

NATBLD: Create Physical and Logical Files

```
PGM
DCL        VAR(&NUMWHSE) TYPE(*DEC) LEN(4 0)
DCL        VAR(&ORDSIZ) TYPE(*DEC) LEN(9 0)
DCL        VAR(&ORDLINSIZ) TYPE(*DEC) LEN(10 0)
DCL        VAR(&NEWORDSIZ) TYPE(*DEC) LEN(9 0)
DCL        VAR(&DLVRYSIZ) TYPE(*DEC) LEN(7 0)
DCL        VAR(&DATE) TYPE(*CHAR) LEN(6)
DCL        VAR(&TPCCDBLIB) TYPE(*CHAR) LEN(10)
DCL        VAR(&TPCCSRCLIB) TYPE(*CHAR) LEN(10)
DCL        VAR(&DDSSRC)     TYPE(*CHAR) LEN(10) +
             VALUE(QDDSSRC)
DCL        VAR(&TPCCPGMLIB) TYPE(*CHAR) LEN(10)
DCL        VAR(&VERSION)    TYPE(*CHAR) LEN(2)
RTVDTAARA  DTAARA(&TPCCPGMLIB/TPCCDBLIB) +
             RTNVAR(&TPCCDBLIB)
RTVDTAARA  DTAARA(&TPCCDBLIB/TPCCWRHS) RTNVAR(&NUMWHSE)
RTVDTAARA  DTAARA(&TPCCDBLIB/TPCCDBVRS) RTNVAR(&VERSION)
RTVDTAARA  DTAARA(&TPCCSRCLIB/TPCCDBLIB) +
             RTNVAR(&TPCCDBLIB)
CHGVAR     VAR(&ORDSIZ) VALUE(&NUMWHSE * 36000)
CHGVAR     VAR(&ORDLINSIZ) VALUE(&NUMWHSE * 360000)
IF         COND(&ORDLINSIZ > 2147483646) THEN(CHGVAR +
             VAR(&ORDLINSIZ) VALUE(2147483646))
CHGVAR     VAR(&NEWORDSIZ) VALUE(&NUMWHSE * 11000)
CHGVAR     VAR(&DLVRYSIZ) VALUE(&NUMWHSE * 1000)
SNDPGMMSG  MSGID(CPF9898) MSGF(QCPFMSG) +
             MSGDTA('Creating Data File Library ' *CAT +
              &TPCCDBLIB) TOPGMQ(*EXT) MSGTYPE(*STATUS)
RTVSYSVAL  SYSVAL(QDATE) RTNVAR(&DATE)
CRTLIB     LIB(&TPCCDBLIB) TEXT('TPC-C Data Files Built ' +
             *CAT (&DATE))
MONMSG     MSGID(CPF0000)
ADDLIBLE   LIB(&TPCCDBLIB)
MONMSG     MSGID(CPF2103) EXEC(DO)
RMVLIBLE   LIB(&TPCCDBLIB)
ADDLIBLE   LIB(&TPCCDBLIB)
ENDDO
SNDPGMMSG  MSGID(CPF9898) MSGF(QCPFMSG) +
             MSGDTA('Creating the Physical Files.') +
             TOPGMQ(*EXT) MSGTYPE(*STATUS)
CHKOBJ     OBJ(&TPCCSRCLIB/MAINICFF) OBJTYPE(*FILE)
CHKOBJ     OBJ(&TPCCSRCLIB/DLVRICFF) OBJTYPE(*FILE)
CHKOBJ     OBJ(&TPCCSRCLIB/DLVRICFF) OBJTYPE(*FILE)

DLTF       FILE(&TPCCSRCLIB/MAINICFF)
DLTF       FILE(&TPCCSRCLIB/DLVRICFF)
DLTF       FILE(&TPCCSRCLIB/STOCICFF)

CRTICFF    FILE(&TPCCSRCLIB/MAINICFF) +
             SRCFILE(&TPCCSRCLIB/QDDSSRC) +
             ACQPGMDEV(PGMDEVR) LVLCHK(*NO)
ADDICFDEVE FILE(&TPCCSRCLIB/MAINICFF) PGMDEV(PGMDEVR) +
             RMTLOCNAME(*REQUESTER)

CRTICFF    FILE(&TPCCSRCLIB/NEWOICFF) +
             SRCFILE(&TPCCSRCLIB/QDDSSRC) +
             ACQPGMDEV(PGMDEVR) LVLCHK(*NO)
ADDICFDEVE FILE(&TPCCSRCLIB/NEWOICFF) PGMDEV(PGMDEVR) +
             RMTLOCNAME(*REQUESTER)

CRTICFF    FILE(&TPCCSRCLIB/PAYMICFF) +
             SRCFILE(&TPCCSRCLIB/QDDSSRC) +
             ACQPGMDEV(PGMDEVR) LVLCHK(*NO)
ADDICFDEVE FILE(&TPCCSRCLIB/PAYMICFF) PGMDEV(PGMDEVR) +
             RMTLOCNAME(*REQUESTER)

CRTICFF    FILE(&TPCCSRCLIB/ORDSICFF) +
             SRCFILE(&TPCCSRCLIB/QDDSSRC) +
             ACQPGMDEV(PGMDEVR) LVLCHK(*NO)
ADDICFDEVE FILE(&TPCCSRCLIB/ORDSICFF) PGMDEV(PGMDEVR) +
             RMTLOCNAME(*REQUESTER)

CRTICFF    FILE(&TPCCSRCLIB/DLVRICFF) +
             SRCFILE(&TPCCSRCLIB/QDDSSRC) +
             ACQPGMDEV(PGMDEVR) LVLCHK(*NO)
ADDICFDEVE FILE(&TPCCSRCLIB/DLVRICFF) PGMDEV(PGMDEVR) +
             RMTLOCNAME(*REQUESTER)

CRTICFF    FILE(&TPCCSRCLIB/STOCICFF) +
             SRCFILE(&TPCCSRCLIB/QDDSSRC) +
             ACQPGMDEV(PGMDEVR) LVLCHK(*NO)
ADDICFDEVE FILE(&TPCCSRCLIB/STOCICFF) PGMDEV(PGMDEVR) +
             RMTLOCNAME(*REQUESTER)
IF  COND(&VERSION *NE '30') THEN( +
DO)
  IF  COND(&VERSION *EQ '31') THEN( +
  CHGVAR VAR(&DDSSRC) VALUE(QDDSSRCV31))
  ELSE CMD( +
  DO)
    IF  COND(&VERSION *EQ '32') THEN( +
      CHGVAR VAR(&DDSSRC) VALUE(QDDSSRCV32))
    ELSE CMD( +
      CHGVAR VAR(&DDSSRC) VALUE(QDDSSRCV40))
  ENDDO
ENDDO

CRTPF      FILE(&TPCCDBLIB/AREFFIL) +
             SRCFILE(&TPCCSRCLIB/&DDSSRC) WAITRCD(*NOMAX)  +
             LVLCHK(*NO)

CRTPF      FILE(&TPCCDBLIB/NEWORDPF) +
             SRCFILE(&TPCCSRCLIB/QDDSSRC) SIZE(&NEWORDSIZ  +
             30000 30000) ALLOCATE(*YES) +
             WAITRCD(*NOMAX) SHARE(*YES) LVLCHK(*NO)

CRTPF      FILE(&TPCCDBLIB/WRHS) SRCFILE(&TPCCSRCLIB/QDDSSRC) +
             SIZE(*NOMAX) WAITRCD(*NOMAX) SHARE(*YES) +
             LVLCHK(*NO)

CRTPF      FILE(&TPCCDBLIB/STOCKPF) +
             SRCFILE(&TPCCSRCLIB/QDDSSRC) SIZE(*NOMAX) +
             WAITRCD(*NOMAX) SHARE(*YES) LVLCHK(*NO)

CRTPF      FILE(&TPCCDBLIB/ORDLINPF) +
             SRCFILE(&TPCCSRCLIB/&DDSSRC) +
             SIZE(&ORDLINSIZ 30000 25000) +
             ALLOCATE(*YES) WAITRCD(*NOMAX) +
             SHARE(*YES) LVLCHK(*NO)

CRTPF      FILE(&TPCCDBLIB/ITEM) SRCFILE(&TPCCSRCLIB/QDDSSRC) +
             SIZE(*NOMAX) WAITRCD(*NOMAX) SHARE(*YES) +
             LVLCHK(*NO)

CRTPF      FILE(&TPCCDBLIB/HSTRY) SRCFILE(&TPCCSRCLIB/&DDSSRC) +
             SIZE(&ORDSIZ 30000 30000) ALLOCATE(*YES) +
             WAITRCD(*NOMAX) SHARE(*YES) LVLCHK(*NO)

CRTPF      FILE(&TPCCDBLIB/DSTRCT) +
             SRCFILE(&TPCCSRCLIB/QDDSSRC) SIZE(*NOMAX) +
             WAITRCD(*NOMAX) SHARE(*YES) LVLCHK(*NO)

CRTPF      FILE(&TPCCDBLIB/CSTMRPF) +
             SRCFILE(&TPCCSRCLIB/&DDSSRC) SIZE(*NOMAX) +
             WAITRCD(*NOMAX) SHARE(*YES) LVLCHK(*NO)

CRTPF      FILE(&TPCCDBLIB/ORDERSPF) +
             SRCFILE(&TPCCSRCLIB/&DDSSRC) SIZE(&ORDSIZ +
             30000 30000) ALLOCATE(*YES) +
             WAITRCD(*NOMAX) SHARE(*YES) LVLCHK(*NO)

CRTPF      FILE(&TPCCDBLIB/DLVRYLOG) +
             SRCFILE(&TPCCSRCLIB/&DDSSRC) SIZE(&DLVRYSIZ +
             10000 10000) ALLOCATE(*YES) +
             WAITRCD(*NOMAX) SHARE(*YES) LVLCHK(*NO)

CRTPF      FILE(&TPCCDBLIB/DLVRYLOGA) +
             SRCFILE(&TPCCSRCLIB/QDDSSRC) SIZE(*NOMAX) +
             WAITRCD(*NOMAX) SHARE(*YES) LVLCHK(*NO)


CRTPRTF    FILE(&TPCCDBLIB/DBUGPRT) +
             SRCFILE(&TPCCSRCLIB/&DDSSRC) LVLCHK(*NO)

CRTPRTF    FILE(&TPCCDBLIB/DBUGPRT2) +
             SRCFILE(&TPCCSRCLIB/&DDSSRC) LVLCHK(*NO)

SNDPGMMSG  MSGID(CPF9898) MSGF(QCPFMSG) +
             MSGDTA('Physical Files created.') +
             TOPGMQ(*EXT) MSGTYPE(*STATUS)


ENDPGM
```

CRTHASH: Create Hashes

```
PGM        PARM(&DATLIB &BLDLIB)
DCL        VAR(&DATLIB) TYPE(*CHAR) LEN(10)
DCL        VAR(&BLDLIB) TYPE(*CHAR) LEN(10)

MONMSG CPF0000
ADDLIBLE   LIB(&BLDLIB)
MONMSG     MSGID(CPF2103)
ADDLIBLE   LIB(&DATLIB)
MONMSG     MSGID(CPF2103)
DLTF       FILE(QGPL/HASHES)
MONMSG     MSGID(CPF0000)
DLTPGM     PGM(QGPL/CSTMR)
DLTPGM     PGM(QGPL/ITEMLF)
DLTPGM     PGM(QGPL/STOCK)
DLTUSRSPC  USRSPC(QGPL/HASH)
CRTSRCPF   FILE(QGPL/HASHES) MBR(HASHES)
CRTCMOD    MODULE(&BLDLIB/C_CREATE) +
             SRCFILE(&BLDLIB/QCSRC) OUTPUT(*PRINT) +
             OPTIMIZE(*FULL) DBGVIEW(*SOURCE)
CRTPGM     PGM(&BLDLIB/C_CREATE) +
             MODULE(&BLDLIB/C_CREATE) +
             BNDSRVPGM(QSYS/QDBCRTHA) ACTGRP(*CALLER)
CRTCMOD    MODULE(&BLDLIB/C_CREATE2) +
             SRCFILE(&BLDLIB/QCSRC) OUTPUT(*PRINT) +
             OPTIMIZE(*FULL) DBGVIEW(*SOURCE)
CRTPGM     PGM(&BLDLIB/C_CREATE2) +
             MODULE(&BLDLIB/C_CREATE2) +
             BNDSRVPGM(QSYS/QDBCRTHA) ACTGRP(*CALLER)
CRTCMOD    MODULE(&BLDLIB/C_CREATE3) +
             SRCFILE(&BLDLIB/QCSRC) OUTPUT(*PRINT) +
             OPTIMIZE(*FULL) DBGVIEW(*SOURCE)
CRTPGM     PGM(&BLDLIB/C_CREATE3) +
             MODULE(&BLDLIB/C_CREATE3) +
             BNDSRVPGM(QSYS/QDBCRTHA) ACTGRP(*CALLER)
SBMJOB     CMD(CALL PGM(&BLDLIB/C_CREATE)) JOBQ(QCTL) +
             MSGQ(&BLDLIB/TPCCBLDMSG)
DLYJOB 5
RCVMSG     MSGQ(&BLDLIB/TPCCBLDMSG)
SBMJOB     CMD(CALL PGM(&BLDLIB/C_CREATE2)) JOBQ(QCTL) +
             MSGQ(&BLDLIB/TPCCBLDMSG)
DLYJOB 5
RCVMSG     MSGQ(&BLDLIB/TPCCBLDMSG)
SBMJOB     CMD(CALL PGM(&BLDLIB/C_CREATE3)) JOBQ(QCTL) +
             MSGQ(&BLDLIB/TPCCBLDMSG)
DLYJOB 5
RCVMSG     MSGQ(&BLDLIB/TPCCBLDMSG)

CLRLIB QRPLOBJ

ENDPGM
```

DLTLOGICLS: Delete Logical Files

```
PGM        PARM(&DATLIB)
DCL        VAR(&DATLIB) TYPE(*CHAR) LEN(10)
MONMSG     MSGID(CPF2105)

DLTF       FILE(&DATLIB/CSTMR)
DLTF       FILE(&DATLIB/CSTMRLFCRT)
DLTF       FILE(&DATLIB/CSTMRLFNAM)
DLTF       FILE(&DATLIB/NEWORD)
DLTF       FILE(&DATLIB/NEWORDLF)
DLTF       FILE(&DATLIB/ORDERS)
DLTF       FILE(&DATLIB/ORDERSLF)
DLTF       FILE(&DATLIB/ORDLIN)
DLTF       FILE(&DATLIB/ORDLINLF)
DLTF       FILE(&DATLIB/STOCK)
DLTF       FILE(&DATLIB/STOCKLF)
```

LOADTPCCF: Database Population Control Program

```
INITTPCCF:  PGM        PARM(&DATALIB &OBJLIB &STRWHCHAR &WRHSCHAR)
            DCL        VAR(&DATALIB) TYPE(*CHAR) LEN(10)
            DCL        VAR(&OBJLIB) TYPE(*CHAR) LEN(10)
            DCL        VAR(&WRHSCHAR) TYPE(*CHAR) LEN(4)
            DCL        VAR(&STRWHCHAR) TYPE(*CHAR) LEN(4)
            DCL        VAR(&STRWRHS5) TYPE(*CHAR) LEN(5)
            DCL        VAR(&ENDWRHS5) TYPE(*CHAR) LEN(5)
            DCL        VAR(&NULLCH) TYPE(*CHAR) LEN(1) VALUE(X'00')

 /* This program submits jobs to call the C                        */
 /* programs that will fill the data base files.                   */
 /* The programs it submits are:                                   */
 /*   LOADW_D    - A C program that fills WRHS and DSTRCT files.   */
 /*   LOADITEM   - A C program that fills the ITEM & STOCK files.  */
 /*   LOADCST    - A C program that fills CSTMR and HSTRY files.   */
 /*   LOADORD    - A C program that fills ORDERS, ORDLIN & NEWORD. */
 /* This program does the overrides that determine how many        */
 /* records to write on each physical I/O.  Messages are sent to   */
 /* the display so the user can track the progress of the build.   */

            CRTMSGQ    MSGQ(&DATALIB/TPCCBLDMSG)
            MONMSG     MSGID(CPF2112)

            OVRDBF     FILE(WRHS) TOFILE(&DATALIB/WRHS) SHARE(*NO) +
                         SEQONLY(*YES 500) /* Blocking set to 500 +
                         records */
            OVRDBF     FILE(DSTRCT) TOFILE(&DATALIB/DSTRCT) +
                         SEQONLY(*YES 500) /* Blocking set to 500 +
                         records */
            OVRDBF     FILE(ITEM) TOFILE(&DATALIB/ITEM) +
                         SEQONLY(*YES 5000) /* Blocking set to +
                         5000 records */
            OVRDBF     FILE(STOCKPF) TOFILE(&DATALIB/STOCKPF) +
                         SEQONLY(*YES 5000) /* Blocking set to +
                         5000 records */
            OVRDBF     FILE(CSTMRPF) TOFILE(DUMMY/CSTMRPF) +
                         SEQONLY(*YES 1000) /* Blocking set to +
                         1000 records */
            OVRDBF     FILE(HSTRY) TOFILE(&DATALIB/HSTRY) +
                         SEQONLY(*YES 500) /* Blocking set to 500 +
                         records */
            OVRDBF     FILE(ORDERSPF) TOFILE(&DATALIB/ORDERSPF) +
                         SEQONLY(*YES 2000) /* Blocking set to +
                         2000 records */
            OVRDBF     FILE(NEWORDPF) TOFILE(&DATALIB/NEWORDPF) +
                         SEQONLY(*YES 2000) /* Blocking set to +
                         2000 records */
            OVRDBF     FILE(ORDLINPF) TOFILE(&DATALIB/ORDLINPF) +
                         SEQONLY(*YES 10000) /* Blocking set to +
                         10000 records */

            CHGVAR     VAR(&STRWRHS5) VALUE(&STRWHCHAR *CAT &NULLCH)
            CHGVAR     VAR(&ENDWRHS5) VALUE(&WRHSCHAR *CAT &NULLCH)

/* Submit the job to call LOADW_D to fill WRHS and DSTRCT files    */

            SBMJOB     CMD(CALL PGM(&OBJLIB/LOADW_D) PARM(&STRWRHS5 +
                         &ENDWRHS5)) JOB(WDJOB) +
                         JOBQ(&OBJLIB/TPCBLDJOBQ) +
                         MSGQ(&DATALIB/TPCCBLDMSG)

/* Submit the job to call LOADITEM to fill ITEM and STOCK files    */

            IF         COND(&STRWHCHAR *EQ '0001') THEN(DO)
            SBMJOB     CMD(CALL PGM(&OBJLIB/LOADITEM) PARM(&STRWRHS5 +
                         &ENDWRHS5)) JOB(ISJOB) +
                         JOBQ(&OBJLIB/TPCBLDJOBQ) +
                         MSGQ(&DATALIB/TPCCBLDMSG)
            ENDDO

/* Submit the job to call LOADCST to fill CSTMR and HSTRY files    */

            SBMJOB     CMD(CALL PGM(&OBJLIB/LOADCST) PARM(&STRWRHS5 +
                         &ENDWRHS5)) JOB(CHJOB) +
                         JOBQ(&OBJLIB/TPCBLDJOBQ) +
                         MSGQ(&DATALIB/TPCCBLDMSG)

/* Submit the job to call LOADORD to fill ORDERS, ORDLIN & NEWORD  */

            SBMJOB     CMD(CALL PGM(&OBJLIB/LOADORD) PARM(&STRWRHS5 +
                         &ENDWRHS5)) JOB(OONJOB) +
                         JOBQ(&OBJLIB/TPCBLDJOBQ) +
                         MSGQ(&DATALIB/TPCCBLDMSG)

/* Wait for the files to be populated                              */

            RCVMSG     MSGQ(&DATALIB/TPCCBLDMSG) WAIT(*MAX)
            RCVMSG     MSGQ(&DATALIB/TPCCBLDMSG) WAIT(*MAX)
            RCVMSG     MSGQ(&DATALIB/TPCCBLDMSG) WAIT(*MAX)
            RCVMSG     MSGQ(&DATALIB/TPCCBLDMSG) WAIT(*MAX)

/* Delete the overrides                                            */

            DLTOVR     FILE(WRHS)
            DLTOVR     FILE(DSTRCT)
            DLTOVR     FILE(ITEM)
            DLTOVR     FILE(STOCKPF)
            DLTOVR     FILE(ORDERSPF)
            DLTOVR     FILE(CSTMRPF)
            DLTOVR     FILE(HSTRY)
            DLTOVR     FILE(ORDLINPF)
            DLTOVR     FILE(NEWORDPF)



            ENDPGM
```

LOADW_D: Fill WRHS and DSTRCT files

```
/* This is the C program that fills the district file (DSTRCT).     */
  /*  */
/* Start include files */
#include <stdlib.h>
#include <errno.h>
#include <recio.h>
#include <string.h>
#include <xxfdbk.h>
#include <milib.h>
#include <micomput.h>
#include <decimal.h>
#include <time.h>
 /* */
#define RANDOM(a,b) ((rand() % (b-a+1))+a)
 /* */
/* MAPINC generates structures for the database files          */
#pragma mapinc("dstrct","*LIBL/dstrct(*all)","both","d _P",,"tpcc")
#include "dstrct"
#pragma mapinc("wrhs","*LIBL/wrhs(*all)","both", "d _P",,"tpcc")
#include "wrhs"
 /* */
  /* Prototypes */
 void LoadWare(int);
 void LoadDstrt(int, int);
 void CreateStr(int, int, char *);
 void CreateZip(char *);
 /* */
  _RFILE
         *Dstrct_File,
         *Wrhs_File;

 tpcc_DSRCD_both_t DRec;
 tpcc_WRRCD_both_t WRec;
  /* */
 static unsigned long int rand_next = 1; /* used in MaxRand routine */
 static char chAlpha[62] = "abcdefghijklmnopqrstuvwxyz0123456789";
 static char chUpAlpha[27] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
 static char tempstr[81];

   /* there are 100 entries each of length 20 */
 char  *CITY_ARR [] =
   {  "Saint_Paul          ", "Saint_Louis         ",   "Concord_Grapes      ",
      "Trenton_New_Jersey  ",  "Bismark_Doughnut    ",   "Cheyenne_Wyoming    ",
      "Juneau_Alaska       ",  "Honolulu_Hawaii     ",   "Phoenix_The_Big_Bird",
      "Topeka_Kansas       ",  "Montgomery          ",   "Elmwood_Illinois    ",
      "Madison_Dolly       ",  "Lansing_Michigan    ",   "Frankfurt_Germany   ",
      "Des_Moines_Iowa     ",
      "Lincoln_Abraham     ",
      "Pierre_South_Dakota ",
      "Dover_Cliffs        ",
      "Helena_Montana      ",
      "San_Antonio         ",
      "Saint_Petersburg    ",
      "Caribu_Maine        ",
      "Sault_Saint_Marie   ",
      "Los_Angeles         ",
      "Minneapolis_St_Paul ",
      "Dallas_Fort_Worth   ",
      "Kalamazoo_Michigan  ",
      "The_Land_Of_Oz      ",
      "Lexington_Kentucky  ",
      "Washington          ",
      "District_Of_Columbia",
      "New_Orleans         ",
      "Boise_Idaho         ",
      "Denver_Colorado     ",
      "Kensington_Maryland ",
      "Riverside_Ca        ",
      "Blue_Mountain       ",
      "New_York_City       ",
      "Nashville_Tennessee ",
      "Fayetteville        ",
      "Berkeley_California ",
      "Maryville_Missouri  ",
      "San_Francisco       ",
      "Indianapolis        ",
      "Saint_Joseph_Mo     ",
      "Rochester_Minnesota ",
      "Fort_Worth          ",
      "Holland_Michigan    ",
      "South_Orange_NJ     ",
      "Stillwater          ",
      "Eugene_Oregon       ",
      "Alexandria_Virginia ",
      "Salt_Lake_City      ",
      "Manitowoc_Wisconsin ",
      "Philadelphia        ",
      "Sioux_Falls         ",
      "West_Lafayette      ",
      "New_Brunswick_NJ    ",
      "Rochester_New_York  ",
      "Oakland_Ca          ",
      "Las_Vegas_Nevada    ",
      "Birmingham_Alabama  ",
      "Arkadelphia_Arkansas",
      "San_Juan_Puerto_Rico",
      "Omaha_Nebraska      ",
      "Walla_Walla         ",
      "Baraboo_Wisconsin   ",
      "Atlanta_Georgia     ",
      "Grand_Forks_ND      ",
      "Tempe_Arizona       ",
      "Knoxville_Tn        ",
      "Fort_Leavenworth    ",
      "Boston_Massachusetts",
      "Danville_Virginia   ",
      "Baltimore_Maryland  ",
      "New_Haven_Ct        ",
      "Claremont_California",
      "Otsego_Michigan     ",
      "Providence          ",
      "Jacksonville        ",
      "Columbia_SC         ",
      "London_England      ",
      "Paris_France        ",
      "Chicago_Illinois    ",
      "Albuquerque         ",
      "Raleigh_NC          ",
      "Kansas_City_Missouri",
      "Tacoma_Washington   ",
      "Oronoco_Mn          ",
      "Charleston_WV       ",
      "Newark_Delaware     ",
      "Burlington_Vermont  ",
      "Damariscotta_Maine  ",
      "Colorado_Springs_Co ",
      "Nacogdoches_Texas   ",
      "Barbourville_Ky     ",
      "North_Carolina      ",
      "Mary_Of_The_Woods   ",
      "Heston_And_Isleworth"    };

    /* There are 100 entries each of length 2.    */
 char *state_arr[] =
    {  "AL", "AK", "AR", "AZ", "CA", "CO", "CT", "DE", "FL", "GA",
       "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD",
       "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NM",
       "NJ", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC",
       "SD", "TN", "TX", "UT", "VT", "VA", "WA", "WV", "WI", "WY",
```

```
    "AC", "AG", "AI", "AM", "AP", "AV", "AW", "CN", "CS", "DI",
    "DM", "DN", "DS", "ED", "EM", "EN", "HN", "HO", "IH", "IM",
    "IR", "IW", "JN", "KA", "KO", "LF", "LI", "NI", "NT", "NZ",
    "OC", "OM", "ON", "PR", "RA", "RO", "SK", "SM", "TC", "TM",
    "TY", "TV", "VI", "VN", "VW", "XT", "YK", "YN", "YW", "ZA"    };
/* */
int main(argc, argv)
register int argc; char **argv;
{
  short int i;
  short int iStrtWrhs = 1;
  short int iEndWrhs = 1;
  short int iCurWrhs = 1;
  short int iDist;
  int iRetCode = 0;
    /* Set up random number generator */
  time_t ltime;
  time(&ltime);
  srand(ltime);          /* seed the random num generator */
  rand_next = (unsigned long int)ltime;
      /* set up alpha seed for random chars */
  strcat(chAlpha, chUpAlpha);
    /* parse the input parms */
  for (argc--, argv++, i=0; argc>0; ++i, argc--, argv++)
  {
    if (i==0) {
      iStrtWrhs = atoi(*argv);
    }
    else {
      iEndWrhs = atoi(*argv);
    }
  }  /* end of parsing input parms */

       /*  */
  if( (Wrhs_File = _Ropen("wrhs","rr+")) == NULL)
  {
    printf("Open of Wrhs file failed\n");
    exit(99);
  }
  if( (Dstrct_File = _Ropen("dstrct","rr+")) == NULL)
  {
    printf("Open of District file failed\n");
    exit(99);
  }

  for (iCurWrhs=iStrtWrhs; iCurWrhs<=iEndWrhs; iCurWrhs++)
  {
    LoadWare(iCurWrhs);
    for (iDist=1; iDist<=10; iDist++)
    {
      LoadDstrt(iCurWrhs, iDist);
    }
  } /* of for (iCurWrhs=iStrtWrhs; iCurWrhs<=iEndWrhs; iCurWrhs++) */

 _Rclose(Dstrct_File);
 _Rclose(Wrhs_File);
 return(iRetCode);
} /* end of main */

/*********************************************************************/
/*   L o a d W a r e                                                 */
/*      Create & write warehouse rec.                                */
/*                                                                   */
/*  PARMS:                                                           */
/*     iWid => current warehouse id                                  */
/*********************************************************************/
void LoadWare(iWid)
  short iWid;
{
  char curwhs[05] = "0001";
  char waddr1[21] = "1";
  char waddr2[21] = "Bldg 1";
  decimal(5,4) d;
  int iX;

  WRec.WID = iWid;
  sprintf(curwhs, "%04d", iWid);
      /* build wname */
  CreateStr(1, 5, tempstr);
  strcat(tempstr, curwhs);
  strcat(tempstr, "W");
  memset(WRec.WNAME, ' ', 10);
  strncpy(WRec.WNAME, tempstr, strlen(tempstr));
      /* build address1 */
  CreateStr(1, 10, waddr1);
  memset(tempstr, '\0', strlen(tempstr));
  strcat(tempstr,"1");
  strcat(tempstr, curwhs);
  strcat(tempstr, " Ave.");
  strcat(tempstr, waddr1);
  memset(WRec.WADDR1, ' ', 20);
  strncpy(WRec.WADDR1, tempstr, strlen(tempstr));
      /* build address2 */
  CreateStr(1, 10, waddr2);
  memset(tempstr, '\0', strlen(tempstr));
  strcat(tempstr,"Bldg 1");
  strcat(tempstr, curwhs);
  strcat(tempstr, waddr2);
  memset(WRec.WADDR2, ' ', 20);
  strncpy(WRec.WADDR2, tempstr, strlen(tempstr));
      /* build city */
  iX = RANDOM(0,99);
  memset(WRec.WCITY, ' ', 20);
  strncpy(WRec.WCITY, CITY_ARR[iX], strlen(CITY_ARR[0]));
      /* build state */
  strncpy(WRec.WSTATE, state_arr[iX], strlen(state_arr[0]));
      /* build zip code */
  CreateZip(tempstr);
  strncpy(WRec.WZIP, tempstr, strlen(tempstr));
      /* build tax */
  d = (RANDOM(0,2000))/10000.0;
  WRec.WTAX = d;
      /* build YTD balance */
  WRec.WYTD = 300000.00;

  _Rwrite(Wrhs_File,(void *)&WRec, sizeof(WRec));
  return;
} /* end of LoadWare */



/*********************************************************************/
/*   L o a d D s t r t                                               */
/*      Create & write district rec.                                 */
/*                                                                   */
/*  PARMS:                                                           */
/*     iWid => current warehouse id                                  */
/*     uchDid => current district id                                 */
/*********************************************************************/
void LoadDstrt(iWid, iCurDist)
  short int iWid;
  short int iCurDist;
{
  char curwhs[05] = "0001";
  char curdist[03] = "01";
  char daddr[21] = "1";
  char tstr[21];
  decimal(5,4) d;
  int iX;

  DRec.DWID = iWid;
  DRec.DID = iCurDist;
  sprintf(curwhs, "%04d", iWid);
  sprintf(curdist, "%02d", iCurDist);

      /* build dname */
  strcpy(tempstr, "D");
  strcat(tempstr, curwhs);
  strcat(tempstr, curdist);
  CreateStr(1, 4, tstr);
  strcat(tempstr, tstr);
  memset(DRec.DNAME, ' ', 10);
  strncpy(DRec.DNAME, tempstr, strlen(tempstr));
      /* build address1 */
  strcpy(daddr, curdist);
  iX = RANDOM(1,999);
  sprintf(tstr, "%03d", iX);
  strcat(daddr, tstr);
  strcat(daddr, " Ave.");
  CreateStr(1, 10, tempstr);
  strcat(daddr, tempstr);
  memset(DRec.DADDR1, ' ', 20);
  strncpy(DRec.DADDR1, daddr, strlen(daddr));
      /* build address2 */
  strcpy(daddr, "Bldg ");
  strcat(daddr, curdist);
  strcat(daddr, " ");
  CreateStr(1, 12, tempstr);
  strcat(daddr, tempstr);
  memset(DRec.DADDR2, ' ', 20);
  strncpy(DRec.DADDR2, daddr, strlen(daddr));
      /* build city */
  iX = RANDOM(0,99);
  memset(DRec.DCITY, ' ', 20);
  strncpy(DRec.DCITY, CITY_ARR[iX], strlen(CITY_ARR[0]));
      /* build state */
  strncpy(DRec.DSTATE, state_arr[iX], strlen(state_arr[0]));
      /* build zip code */
  CreateZip(tstr);
  strncpy(DRec.DZIP, tstr, strlen(tstr));

      /* build tax */
  d = (RANDOM(0,2000)) /10000.0;
  DRec.DTAX = d;
      /* build YTD balance */
  DRec.DYTD = 30000.00;
      /* build next order id */
  DRec.DNXTOR = 3001;

  _Rwrite(Dstrct_File,(void *)&DRec,  sizeof(DRec));
  return;
} /* end of LoadDstrt */


/*********************************************************************/
/*   C r e a t e S t r                                               */
/*      Create string of random alphanumeric characters.             */
/*                                                                   */
/*  PARMS:                                                           */
/*     iMin => minimum length of string                              */
/*     iMax => maximum length of string                              */
/*     szTemp => address to store created string                     */
/*  If string is not variable length, then set iMin and iMax to     */
/*  the actual fixed length.                                         */
/*********************************************************************/
void CreateStr(iMin, iMax, szTemp)
  int iMin, iMax;
  char *szTemp;
{
  int iAlphalen,
      i, j;

  iAlphalen = RANDOM(iMin,iMax);   /* determine str len  */
  memset(szTemp, '\0', strlen(szTemp));
  for (j=0; j<iAlphalen; j++)
  {
    i = RANDOM(0,61);
    szTemp[j]= chAlpha[i];
  }
  return;
} /* end of CreateStr */

/*********************************************************************/
/*   C r e a t e Z i p                                               */
/*      Create zip code string.                                      */
/*                                                                   */
/*  PARMS:                                                           */
/*    Returns zip code string.                                       */
/*********************************************************************/
void CreateZip(zipstr)
char *zipstr;
{
  int iZ;

  iZ = RANDOM(0,99999);
  sprintf(zipstr, "%05d", iZ);
  strcat(zipstr, "1111");
  return;
} /* end of CreateZip */


  /* Redefine max generated random num. Default is 32,767. */
#undef RAND_MAX
#define RAND_MAX 99999
 /*********************************************************************/
 /*   M a x R a n d                                                   */
 /*      Create random number larger than 32767.                      */
 /*  NOTE: This is the rand() function used by AS/400 with one        */
 /*      slight modification.  In the rand() function, rand_next is   */
 /*      shifted right by 16 rather than the 14 used here.  Shifting  */
 /*      by 16 produces a max value of 65535.                         */
 /*                                                                   */
 /*  PARMS:                                                           */
 /*    Returns zip code string.                                       */
 /*********************************************************************/
int MaxRand(void)
{
  int rand_temp;

  rand_next = rand_next * 1103515245 + 12345;
  rand_temp = (rand_next >> 14)%((unsigned long)(RAND_MAX)+1);
  return(rand_temp);

} /* end of CreateZip */
```

LOADITEM: Fill ITEM and STOCK Files

```
/* This is the C program that fills:                              */
/*   item file (ITEM)                                             */
/*   stock file (STOCK)                                           */
 /*  */
/* Start include files */
#include <stdlib.h>
#include <errno.h>
#include <recio.h>
#include <string.h>
#include <xxfdbk.h>
#include <milib.h>
#include <micomput.h>
#include <decimal.h>
#include <time.h>
 /* */
 #define MAXITEMS 100000
/* #define MAXITEMS 1000 */
#define RANDOM(a,b) ((rand() % (b-a+1))+a)
 /* */
/* MAPINC generates structures for the database files          */
#pragma mapinc("item","*LIBL/item(*all)","both","d _P",,"tpcc")
#include "item"
#pragma mapinc("stockpf","*LIBL/stockpf(*all)","both", "d _P",,"tpcc")
#include "stockpf"
 /* */
  /* Prototypes */
 void LoadItem(void);
 void LoadStock(int);
 void CreateStr(int, int, char *);
 /* */
  _RFILE
        *Item_File,
        *Stock_File;

 tpcc_ITRCD_both_t ItemRec;
 tpcc_STRCD_both_t StkRec;

   short int iStrtWrhs = 1;
   short int iEndWrhs = 1;
   short int iCurWrhs = 1;
 char chAlpha[62] = "abcdefghijklmnopqrstuvwxyz0123456789";
 char chUpAlpha[27] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
 char tempstr[81];

   /* There are 3 sets of characters that are concatenated */
   /* to make the district information fields for Stock */
 char *DistInf1[] = {
   "abcdefghijk", "nowisthetim", "thisiswhere",
   "whatdoyouno", "notmuchthen", "whydoyouask",
   "Idonotknow_", "sowhereisit", "inthecities",
   "andthetowns"};

 char *DistInf2[] = {
   "01", "02", "03", "04", "05", "06", "07", "08", "09", "10"};

 char *DistInf3[] = {
   "mnopqrstuvw", "whydidyoudo", "interactive",
   "batchwork__", "systemview_", "performance",
   "Ineverwork_", "Ialwaysplay", "nexttolast_",
   "nowthefinal"};

   /* There are 100 entries, each of length 24 */
 char *ItemNameStr[]  = {
   "Ball_Point_Pen          ",   "Sailboat_Fuel_Tank      ",
   "Flowers_(Poppies)       ",   "VCR_Road_Warrior_Game   ",
   "Bird_House_Plans        ",   "Pushbutton_Telephone    ",
   "Surgical_Gloves         ",   "1_Gallon_Distilled_Water",
   "Black_Labrador_Puppies  ",   "Plain_Pockets_Pants     ",
   "Electric_Plane          ",   "Childs_Car_Seat         ",
   "Gray_Hooded_Sweatshirts ",   "Lone_Ranger_Costume     ",
   "Jiffy_Pain_Pills        ",   "Devil's_Food_Cake       ",
   "Full_Length_Fox_Fur     ",   "AM_FM_Stereo_Car_Radio  ",
   "Cheddar_Cheese_Ball     ",   "Elephant_Ear_Mushroom   ",
   "White_Traffic_Lane_Paint",   "Plastic_Garbage_Pail    ",
   "Four_Blade_Ceiling_Fans ",   "Doll_House_Furniture    ",
   "Apple_Core_Remover      ",   "Cola_Twelve_Pack        ",
   "Italian_Food_Cook_Book  ",   "Wool_Toe_Socks          ",
   "Peel_Good_Vitamins      ",   "100'_Rubber_Garden_Hose ",
   "Orange_Golf_Balls       ",   "Snorkle_And_Fins_Set    ",
   "IBM_Facsimile_Machine   ",   "5_lb._Bag_Flour         ",
   "Fleece_Lined_Wool_Gloves",   "Bermuda_Grass_Seed      ",
   "Front_And_Rear_Floor_Mat",   "Waiter_Costume          ",
   "Magical_Mystery_Maze    ",   "Red_Safety_Flares       ",
   "Silver_Push_Pins        ",   "QuartzDigital_Wristwatch",
   "10_lb_Test_Fishing_Line ",   "Blue_Pile_Carpeting     ",
   "Red_Brick_House         ",   "Bear_Claw_Doughnuts     ",
   "AS/400_Advanced_Series  ",   "12_Inch_Wooden_Ruler    ",
   "Square_Plastic_Calendar ",   "Twelve_Num_Two_Pencils  ",
   "Pad_Yellow_Legal_Paper  ",   "Change_Machine          ",
   "Street_Hockey_Balls     ",   "Half_Inch_3_Ring_Binders",
   "Sporting_Good_Catalog   ",   "Ten_Gallon_Hats         ",
   "Promotion_Notification  ",   "32_Gal._Keg_Beer        ",
   "Business_Cards          ",   "Pebble_Beach_Poster     ",
   "Texas_Residence_Forms   ",   "COBOL_Programmers_Guide ",
   "25_Inch_Color_TVs       ",   "Black_Magic_Marker      ",
   "Telephone_Books         ",   "Riding_Lawnmower        ",
   "$7500_Gift_Certificate  ",   "Compact_Disc_Player     ",
   "Four_Drawer_Cabinets    ",   "Dry-Erase_Markers       ",
   "Multi-System_Paper      ",   "Message_Waiting_Light   ",
   "Automotive_Care_Manuals ",   "Carton_Ceiling_Tiles    ",
   "12_Ounce_Styrofoam_Cup  ",   "Phone_Calling_Card      ",
   "Inflatable_Globe        ",   "Walnut_Magazine_Rack    ",
   "Over_Under_Shotgun      ",   "While_You_Were_Gone_Memo",
   "Three_Tea_Bags          ",   "Bowling_Pin_Spotter     ",
   "Golf_Club_Cleaner       ",   "Staple_Removers         ",
   "Downtown_Parking_Permit ",   "Sliding_Glass_Door_Locks",
   "Floral_Pattern_Lampshade",   "Junior_College_Books    ",
   "Two_Wheel_Bike          ",   "Zoo_Season_Pass         ",
   "Baseball_Tickets        ",   "Four_Person_Tent        ",
   "Cross_Country_Ski_Set   ",   "Motorcycle_Sidecars     ",
   "Fireplace_Tools         ",   "Radio_Controlled_Plane  ",
   "Oak_Wall_Clock          ",   "Rubber_Baby_Buggy_Wheel ",
   "Anti_Static_Strap       ",   "Old_Wooden_Toothpicks   " };


/* */
int main(argc, argv)
register int argc; char **argv;
{
  short int i;
  int iRetCode = 0;
  int item;

    /* Set up random number generator */
  time_t ltime;
  time(&ltime);
  srand(ltime);          /* seed the random num generator */
     /* set up alpha seed for random chars */
  strcat(chAlpha, chUpAlpha);
   /* parse the input parms */
  for (argc--, argv++, i=0; argc>0; ++i, argc--, argv++)
  {
    if (i==0) {
      iStrtWrhs = atoi(*argv);
    }
    else {
      iEndWrhs = atoi(*argv);
   /* strcpy(endwrhs, *argv);             */
   /* printf("endwrhs = %4s \n", endwrhs); */
    }
  }  /* end of parsing input parms */

       /*  */
  if( (Item_File = _Ropen("item","rr+")) == NULL)
  {
    printf("Open of Item file failed\n");
    exit(99);
  }
  if( (Stock_File = _Ropen("stockpf","rr+")) == NULL)
  {
    printf("Open of Stockpf file failed\n");
    exit(99);
  }
    /*************************/
    /*  Start of true build  */
    /*************************/
  LoadItem();
  for (item=1; item<=MAXITEMS; item++)
  {
    LoadStock(item);
  }

  _Rclose(Item_File);
  _Rclose(Stock_File);

} /* end of main */


/*****************************************************************/
/*   L o a d I t e m                                             */
/*      Create & write item records.                             */
/*                                                               */
/*  PARMS: none                                                  */
/*****************************************************************/
void LoadItem()
{
  decimal(5,2) d;
  int i, iX;

  for (i=1; i<=MAXITEMS; i++)
  {
    ItemRec.IID = i;
      /* build image id 1..10,000 */
    iX = RANDOM(1,10000);
    ItemRec.IMAGEID = iX;

      /* build iname */
    iX = RANDOM(0,99);
    memset(ItemRec.INAME, ' ', 24);
    strncpy(ItemRec.INAME, ItemNameStr[iX],
                           strlen(ItemNameStr[0]));

      /* build IPRICE */
    d =  (RANDOM(100,10000)) / 100.00;
    ItemRec.IPRICE = d;
      /* Build IDATA random string 26 to 50 chars */
    CreateStr(26, 50, tempstr);
    if ((RANDOM(0,9)) == 0)
    {
      iX = (RANDOM(0,(strlen(tempstr)-8)));
      memcpy(&tempstr[iX], "ORIGINAL", 8);
    }
    memset(ItemRec.IDATA, ' ', 50);
    strncpy(ItemRec.IDATA, tempstr, strlen(tempstr));
    _Rwrite(Item_File,(void *)&ItemRec,  sizeof(ItemRec));
  } /* end of for */

  return;
} /* end of LoadItem */


/*****************************************************************/
/*   L o a d S t o c k                                           */
/*      Create & write stock records.                            */
/*                                                               */
/*  PARMS:                                                       */
/*     item => current item id                                   */
/*****************************************************************/
void LoadStock(item)
 int item;
{
  int i, iX;
  char StockDist[25] ;

  for (iCurWrhs=iStrtWrhs; iCurWrhs<=iEndWrhs; iCurWrhs++)
  {
      /* create STIID */
    StkRec.STIID = item;
      /* create STWID */
    StkRec.STWID = iCurWrhs;
      /* create STQTY between 10..100*/
    StkRec.STQTY = RANDOM(10,100);
     /* CREATES STDInn (district info) */
     iX = RANDOM(0, 9);
    memcpy(StockDist, DistInf1[iX], 11);
    memcpy((void *)&StockDist[11], DistInf2[0], 2);
     iX = RANDOM(0, 9);
    memcpy((void *)&StockDist[13], DistInf3[iX], 11);
    strncpy(StkRec.STDI01, StockDist, 24);
     iX = RANDOM(0, 9);
    memcpy(StockDist, DistInf1[iX], 11);
    memcpy((void *)&StockDist[11], DistInf2[1], 2);
     iX = RANDOM(0, 9);
    memcpy((void *)&StockDist[13], DistInf3[iX], 11);
    strncpy(StkRec.STDI02, StockDist, 24);
     iX = RANDOM(0, 9);
    memcpy(StockDist, DistInf1[iX], 11);
    memcpy((void *)&StockDist[11], DistInf2[2], 2);
     iX = RANDOM(0, 9);
    memcpy((void *)&StockDist[13], DistInf3[iX], 11);
    strncpy(StkRec.STDI03, StockDist, 24);
     iX = RANDOM(0, 9);
    memcpy(StockDist, DistInf1[iX], 11);
    memcpy((void *)&StockDist[11], DistInf2[3], 2);
     iX = RANDOM(0, 9);
    memcpy((void *)&StockDist[13], DistInf3[iX], 11);
    strncpy(StkRec.STDI04, StockDist, 24);
     iX = RANDOM(0, 9);
    memcpy(StockDist, DistInf1[iX], 11);
    memcpy((void *)&StockDist[11], DistInf2[4], 2);
```

```
     iX = RANDOM(0, 9);
    memcpy((void *)&StockDist[13], DistInf3[iX], 11);
    strncpy(StkRec.STDI05, StockDist, 24);
     iX = RANDOM(0, 9);
    memcpy(StockDist, DistInf1[iX], 11);
    memcpy((void *)&StockDist[11], DistInf2[5], 2);
     iX = RANDOM(0, 9);
    memcpy((void *)&StockDist[13], DistInf3[iX], 11);
    strncpy(StkRec.STDI06, StockDist, 24);
     iX = RANDOM(0, 9);
    memcpy(StockDist, DistInf1[iX], 11);
    memcpy((void *)&StockDist[11], DistInf2[6], 2);
     iX = RANDOM(0, 9);
    memcpy((void *)&StockDist[13], DistInf3[iX], 11);
    strncpy(StkRec.STDI07, StockDist, 24);
     iX = RANDOM(0, 9);
    memcpy(StockDist, DistInf1[iX], 11);
    memcpy((void *)&StockDist[11], DistInf2[7], 2);
     iX = RANDOM(0, 9);
    memcpy((void *)&StockDist[13], DistInf3[iX], 11);
    strncpy(StkRec.STDI08, StockDist, 24);
     iX = RANDOM(0, 9);
    memcpy(StockDist, DistInf1[iX], 11);
    memcpy((void *)&StockDist[11], DistInf2[8], 2);
     iX = RANDOM(0, 9);
    memcpy((void *)&StockDist[13], DistInf3[iX], 11);
    strncpy(StkRec.STDI09, StockDist, 24);
     iX = RANDOM(0, 9);
    memcpy(StockDist, DistInf1[iX], 11);
    memcpy((void *)&StockDist[11], DistInf2[9], 2);
     iX = RANDOM(0, 9);
    memcpy((void *)&StockDist[13], DistInf3[iX], 11);
    strncpy(StkRec.STDI10, StockDist, 24);
      /* create STDATA random string 26 to 50 chars */
    CreateStr(26, 50, tempstr);
    if ((RANDOM(0,9)) == 0)
    {
      iX = (RANDOM(0,(strlen(tempstr)-8)));
      memcpy(&tempstr[iX], "ORIGINAL", 8);
    }
    memset(StkRec.STDATA, ' ', 50);
    strncpy(StkRec.STDATA, tempstr, strlen(tempstr));

    StkRec.STYTD = 0;
    StkRec.STORDRS = 0;
    StkRec.STREMORD = 0;

    _Rwrite(Stock_File,(void *)&StkRec,  sizeof(StkRec));

  } /* end of for loop */

  return;
} /* end of LoadStock */


/*********************************************************************/
/*   C r e a t e S t r                                               */
/*      Create string of random alphanumeric characters.             */
/*                                                                   */
/*  PARMS:                                                           */
/*     iMin => minimum length of string                              */
/*     iMax => maximum length of string                              */
/*     szTemp => address to store created string                     */
/*  If string is not variable length, then set iMin and iMax to     */
/*  the actual fixed length.                                         */
/*********************************************************************/
void CreateStr(iMin, iMax, szTemp)
  int iMin, iMax;
  char *szTemp;
{
  int iAlphalen,
      i, j;

  iAlphalen = RANDOM(iMin,iMax);    /* determine str len  */
  memset(szTemp, '\0', strlen(szTemp));
  for (j=0; j<iAlphalen; j++)
  {
    i = RANDOM(0,61);
    szTemp[j]= chAlpha[i];
  }
  return;
} /* end of CreateStr */
```

LOADCST: Fill CSTMR and HSTRY Files

```
/*********************************************************************/
/* This program fills the customer and history files.                */
/* It fills them all simultaneously. It fills a customer             */
/* record, then the history record for that customer.                */
/*                                                                   */
/*                                                                   */
/*   C  H  A  N  G  E  S     &      U  P  D  A  T  E  S              */
/*                                                                   */
/*                                                                   */
/*    PROGRAM-ID. LOADCSTORD.                                        */
/*    AUTHOR. PPCOC.                                                 */
/*    INSTALLATION. ROCHESTER.                                       */
/*    DATE-WRITTEN. 03/04/96.                                        */
/*    DATE-COMPILED. 03/06/96.                                       */
/*********************************************************************/

/* Start include files */
#include <stdlib.h>
#include <errno.h>
#include <recio.h>
#include <string.h>
#include <xxfdbk.h>
#include <milib.h>
#include <micomput.h>
#include <decimal.h>
#include <time.h>
  /* */
#define CUSTPERDIST 3000
#define RANDOM(a,b) (rand() % (b-a+1))+a
 /* */
/* MAPINC generates structures for the database files          */
#pragma mapinc("cstmrpf","*libl/cstmrpf(*all)","both", "d _P",,"tpcc")
#include "cstmrpf"
#pragma mapinc("hstry","*libl/hstry(*all)","both", "d _P",,"tpcc")
#include "hstry"
  /*  */
  /* Prototypes */
 void LoadCust(int, int, int);
 void LoadHstry(int, int, int);
 void CreateCLast(int, char *);
int NURand(int, int, int);
void CreateStr(int, int, char *);
void CreateZip(char *);
void GetDateTime(char *, char *);
int MaxRand(void);
 /*  */
static int C=7;
static unsigned long int rand_next = 1; /* used in MaxRand routine */
    /*    */
      /* There are 100 names.       */
 char *szFNameâã =
   {"FrankGifford    ",  "FranTarkenton   ",   "GailStormyNight ",
    "JohnWayne       ", "JaneSeeDickRun  ",   "JeannieWithLight",
    "DavesNotHere    ",  "MaryMartin      ",   "JefferyTheWaiter",
    "CarriePromQueen ",  "StephanKing     ",   "KaraKing        ",
    "AnnieOakley     ",  "BillClements    ",   "FannyFarmer     ",
    "AllanLuden      ",  "BernieceLattitude",  "MelindaFisher   ",
    "ElaineBoosler   ",  "FrankieAvalon   ",   "Francine        ",
    "GaleGordon      ",  "JohnBoyWalton   ",   "JeanneCRiley    ",
    "DavidBowie      ",  "KuklaFranOllie  ",   "PhillipOfWales  ",
    "KerryCWilliams  ",  "Stephanie       ",   "AnnaSueBrighton ",
    "BillyBob        ",  "FayDunawayWith  ",   "PamelaSueMartin ",
    "AlanAlda        ",  "BernardPFife    ",   "LindaDayGeorge  ",
    "RobertConrad    ",  "Franklin        ",   "FrancisTheMule  ",
    "GayleMcDonald   ",  "JanMurray       ",   "JoanneCassidy   ",
    "MarkTheSpot     ",  "Geoffrey        ",   "PhyllisDiller   ",
    "DoogieHowser    ",  "Anastasia       ",   "AlvinChipmunk   ",
    "AlexPKeaton     ",  "BelindaMcWilliam",   "EllynBurnstyne  ",
    "JayePMorgan     ",  "JoeFromKokomo   ",   "JoeyHeatherton  ",
    "Gilligan        ",  "TheSkipper2     ",   "CarryBishop     ",
    "BenjaminFranklin",  "AbrahamLincoln  ",   "GeorgeWBush     ",
    "VincentPrice    ",  "PotsieWeber     ",   "RichieCunningham",
    "Alexander       ",  "PeterTheGreater ",   "MickeyMouse     ",
    "DonaldDuck      ",  "BabyHuey        ",   "CrystalGayle    ",
    "SnidelyWhiplash ",  "HastaLaVista    ",   "GeorgeWashington",
    "KareemAbdulJabar",  "WhitneyHouston  ",   "BobSmith        ",
    "PerryMason      ",  "PaulDrake       ",   "DellaMainStreet ",
    "MissPiggy       ",  "TheCookieMonster",   "KermitTheFrog   ",
    "BluesBrothers   ",  "BeaverCleaver   ",   "Anriqueta       ",
    "Crenshaw        ",  "AndyTaylor      ",   "ClaytonWilliams ",
    "BillClinton     ",  "Elizabeth       ",   "MichaelJordan   ",
    "PaulRevere      ",  "JohnGlenn       ",   "WashingtonCarver",
    "DanJohnson      ",  "BubbaTexan      ",   "GGordonLiddy    ",
    "ScottSimpson    ",  "BenKnight       ",
    "NolanRyan       ",  "AlbertEinstein  "   };

  /*  */
  /* There are 100 cities. */
   char  *CITY_ARRâã =
  {  "Saint_Paul          ",  "Saint_Louis         ",
     "Concord_Grapes      ",  "Trenton_New_Jersey  ",
     "Bismark_Doughnut    ",  "Cheyenne_Wyoming    ",
     "Juneau_Alaska       ",  "Honolulu_Hawaii     ",
     "Phoenix_The_Big_Bird",  "Topeka_Kansas       ",
     "Montgomery          ",  "Elmwood_Illinois    ",
     "Madison_Dolly       ",  "Lansing_Michigan    ",
     "Frankfurt_Germany   ",  "Des_Moines_Iowa     ",
     "Byron_Minnesota     ",  "Pierre_South_Dakota ",
     "Dover_Cliffs        ",  "Helena_Montana      ",
     "San_Antonio         ",  "Saint_Petersburg    ",
     "Caribu_Maine        ",  "Sault_Saint_Marie   ",
     "Los_Angeles         ",  "Minneapolis_St_Paul ",
     "Dallas_Fort_Worth   ",  "Kalamazoo_Michigan  ",
     "The_Land_Of_Oz      ",  "Lexington_Kentucky  ",
     "Washington          ",  "District_Of_Columbia",
     "New_Orleans         ",  "Boise_Idaho         ",
     "Kensington_Maryland ",  "Denver_Colorado     ",
     "Blue_Mountain       ",  "Riverside_Ca        ",
     "New_York_City       ",  "Nashville_Tennessee ",
     "Mayberry_RFD        ",  "Berkeley_California ",
     "Maryville_Missouri  ",  "San_Francisco       ",
     "Indianapolis        ",  "Saint_Joseph_Mo     ",
     "Rochester_Minnesota ",  "Fort_Worth          ",
     "Holland_Michigan    ",  "South_Orange_NJ     ",
     "Stillwater          ",  "Eugene_Oregon       ",
     "Alexandria_Virginia ",  "Salt_Lake_City      ",
     "Manitowoc_Wisconsin ",  "Philadelphia        ",
     "Sioux_Falls         ",  "West_Lafayette      ",
     "New_Brunswick_NJ    ",  "Rochester_New_York  ",
     "Oakland_Ca          ",  "Las_Vegas_Nevada    ",
     "Birmingham_Alabama  ",  "Arkadelphia_Arkansas",
     "San_Juan_Puerto_Rico",  "Omaha_Nebraska      ",
     "Walla_Walla         ",  "Baraboo_Wisconsin   ",
     "Atlanta_Georgia     ",  "Grand_Forks_ND      ",
     "Tempe_Arizona       ",  "Knoxville_Tn        ",
     "Fort_Leavenworth    ",  "Boston_Massachusetts",
     "Danville_Virginia   ",  "Baltimore_Maryland  ",
     "New_Haven_Ct        ",  "Claremont_California",
     "Otsego_Michigan     ",  "Providence          ",
     "Jacksonville        ",  "Columbia_SC         ",
     "London_England      ",  "Paris_France        ",
     "Chicago_Illinois    ",  "Albuquerque         ",
     "Raleigh_NC          ",  "Kansas_City_Missouri",
     "Tacoma_Washington   ",  "Oronoco_Mn          ",
     "Charleston_WV       ",  "Newark_Delaware     ",
     "Burlington_Vermont  ",  "Damariscotta_Maine  ",
     "Colorado_Springs_Co ",  "Nacogdoches_Texas   ",
     "Barbourville_Ky     ",  "Boca_Raton_Fl       ",
     "Mary_Of_The_Woods   ",  "Heston_And_Isleworth"    };

   /*  */
   /*  There are 100 states */
char *state_arrâã =
   {  "AL", "AK", "AR", "AZ", "CA", "CO", "CT", "DE", "FL", "GA",
      "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD",
      "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NM",
      "NJ", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC",
      "SD", "TN", "TX", "UT", "VT", "VA", "WA", "WV", "WI", "WY",
      "AC", "AS", "AI", "AM", "AP", "AV", "AW", "CN", "CS", "DI",
      "DM", "DN", "DS", "ED", "EM", "EN", "HN", "HO", "IH", "IM",
      "IR", "IW", "JN", "KA", "KO", "LF", "LI", "NI", "NT", "NZ",
      "OC", "OM", "ON", "PR", "RA", "RO", "SK", "SM", "TC", "TM",
      "TY", "TV", "VI", "VN", "VW", "XT", "YK", "YN", "YW", "ZA"    };
 /*  */
tpcc_CSRCD_both_t CustRec;
tpcc_HSRCD_both_t HstRec;
 /*  */
static char chAlpha[62] =
      "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";
static char chNumStr[11] = "0123456789";
static char tempstr[81];
static time_t ltime;
static char TimeStr[7];
static char DateStr[9];
 /*  */
  _RFILE
          *Cstmr_File,
          *Hstry_File;
/* */
int main(argc, argv)
register int argc; char **argv;
{
  short int i;
  short int iStrtWrhs = 1;
```

```
  short int iEndWrhs = 1;
  short int iCurWrhs = 1;
  short int iDist;
  int iCurCid;
  int iRetCode = 0;
    /* Set up random number generator */
  time(&ltime);
  srand(ltime);          /* seed the random num generator */
  rand_next = (unsigned long int)ltime;
      /* set up alpha seed for random chars */
  strcat(chAlpha, chNumStr);
    /* parse the input parms */
  for (argc--, argv++, i=0; argc>0; ++i, argc--, argv++)
  {
    if (i==0) {
      iStrtWrhs = atoi(*argv);
    }
    else {
      iEndWrhs = atoi(*argv);
    }
  }  /* end of parsing input parms */

  if( (Hstry_File = _Ropen("hstry","rr+","blkrcd=Y")) == NULL)
  {
    printf("Open of History file failed\n");
    exit(99);
  }
  if( (Cstmr_File = _Ropen("cstmrpf","rr+","blkrcd=Y")) == NULL)
  {
    printf("Open of Customer file failed\n");
    exit(99);
  }


  for (iCurCid=1; iCurCid<=CUSTPERDIST; iCurCid++)
  {
    for (iCurWrhs=iStrtWrhs; iCurWrhs<=iEndWrhs; iCurWrhs++)
    {
      for (iDist=1; iDist<=10; iDist++)
      {
        LoadHstry(iCurCid, iCurWrhs, iDist);
        LoadCust(iCurWrhs, iDist, iCurCid);
      }
    } /* of for (iCurWrhs=iStrtWrhs; iCurWrhs<=iEndWrhs; iCurWrhs++) */
  }

 _Rclose(Cstmr_File);
 _Rclose(Hstry_File);

 return(iRetCode);
} /* end of main */


/*****************************************************************/
/*   L o a d C u s t                                             */
/*      Create & write customer recs.                            */
/*                                                               */
/*  PARMS:                                                       */
/*     iWid => current warehouse id                              */
/*     iDid => current district id                               */
/*****************************************************************/
void LoadCust(iWid, iDid, iCid)
  short int iWid;
  short int iDid;
  int iCid;
{
  char tstr[21];
  int iX;
  static char cdataStr[500];

  CustRec.CWID = iWid;
  CustRec.CDID = iDid;
  CustRec.CID = iCid;

      /* build cfirst. array 0..99 */
  iX = RANDOM(0,99);
  memset(CustRec.CFIRST, ' ', 16);
  if ((iWid==1) && (iDid==1) && (iCid==1))
  {
     sprintf(CustRec.CFIRST,"C_load = %d",C);

  }
  else
  {
  strncpy(CustRec.CFIRST, szFNameâiXã, strlen(szFName[0]));
  }
      /* build cinit. */
  strncpy(CustRec.CINIT, "OE", 2);
      /* build clast */
  if (iCid < 1001)
  {
    CreateCLast((iCid-1), tstr);
  }
  else
  {
    iX = NURand(255, 0, 999);
    CreateCLast((iX), tstr);
  }
  memset(CustRec.CLAST, ' ', 16);
  strncpy(CustRec.CLAST, tstr, strlen(tstr) );
      /* build address1 */
  CreateStr(10, 20, tstr);
  memset(CustRec.CADDR1, ' ', 20);
  strncpy(CustRec.CADDR1, tstr, strlen(tstr));
      /* build address2 */
  CreateStr(10, 20, tstr);
  memset(CustRec.CADDR2, ' ', 20);
  strncpy(CustRec.CADDR2, tstr, strlen(tstr));
      /* build city */
  iX = RANDOM(0,99);
  memset(CustRec.CCITY, ' ', 20);
  strncpy(CustRec.CCITY, CITY_ARR[iX], strlen(CITY_ARR[iX]));
      /* build state */
  strncpy(CustRec.CSTATE, state_arr[iX], 2);
      /* build zip code */
  CreateZip(tstr);
  strncpy(CustRec.CZIP, tstr, strlen(tstr));

      /* build phone # */
  memset(tstr, '\0', strlen(tstr));
  for (iX = 0; iX < 16; iX++)
  {
    CustRec.CPHONE[iX] = chNumStr[(RANDOM(0,9))];
  }
      /* build credit limit */
  CustRec.CCRDLM = 50000.00;
      /* build credit status */
  if ( (RANDOM(0,9)) > 0 ) {
    CustRec.CCREDT[0] = 'G';
  }
  else
    CustRec.CCREDT[0] = 'B';
  CustRec.CCREDT[1] = 'C';
      /* build Cust balance */
  CustRec.CBAL = -10.00;
      /* build Cust YTD payments */
  CustRec.CYTD = 10.00;
      /* build Cust pay count & delivery count */
  CustRec.CPAYCNT = 1;
  CustRec.CDELCNT = 0;
      /* build Cust Discount */
  CustRec.CDCT = ((RANDOM(0,5000)) / 10000.00);
      /* build Cust Data */
  CreateStr(300, 500, cdataStr);
  memset(CustRec.CDATA, ' ', 500);
  strncpy(CustRec.CDATA, cdataStr, strlen(cdataStr));
      /* build Cust Date & Time */
  GetDateTime(TimeStr, DateStr);
  strncpy(CustRec.CLDATE, DateStr, strlen(DateStr));
  strncpy(CustRec.CLTIME, TimeStr, strlen(TimeStr));

  _Rwrite(Cstmr_File,(void *)&CustRec,  sizeof(CustRec));

  return;
} /* end of LoadCust */


/*****************************************************************/
/*   L o a d H s t r y                                           */
/*      Create & write history rec.                              */
/*                                                               */
/*  PARMS:                                                       */
/*     iWid => current warehouse id                              */
/*****************************************************************/
void LoadHstry(iCid, iWid, iDid)
  int iCid;
  short iWid;
  short iDid;
{


  HstRec.HCWID = iWid;
  HstRec.HWID = iWid;
  HstRec.HDID = iDid;
  HstRec.HCDID = iDid;
  HstRec.HCID = iCid;

  HstRec.HAMT = 10.00;
      /* build Hist Data */
  CreateStr(12, 24, tempstr);
  memset(HstRec.HDATA, ' ', 24);
  strncpy(HstRec.HDATA, tempstr, strlen(tempstr));
      /* build Hist Date & Time */
  GetDateTime(TimeStr, DateStr);
  strncpy(HstRec.HTIME, TimeStr, strlen(TimeStr));
  strncpy(HstRec.HDATE, DateStr, strlen(DateStr));

  _Rwrite(Hstry_File,(void *)&HstRec,  sizeof(HstRec));
} /* end of LoadHstry */


/*****************************************************************/
/*   N U R a n d                                                 */
/*      Return integer value from nurand function.               */
/*      function is                                              */
/*  NURand(A,x,y) = (((rand(0,A) | rand(x,y))+C) % (y-x+1)) + x   */
/*     A is constant chosen according to size of range [x..y]    */
/*     exp1 | exp2 stands for bitwise logical OR of exp1 and exp2 */
/*     rand(x,y) stands for randomly selected within âx..yã      */
/*       C is run-time constant randomly chosen within [0..A]    */
/*                                                               */
/*  PARMS:                                                       */
/*     A => integer value                                        */
/*     x => minimum value in range                               */
/*     y => maximum value in range                               */
/*****************************************************************/
int NURand(A, x, y)
  int A;
  int x;
  int y;
{
  int retval;
  int exp1, exp2;

  exp1 = RANDOM(0,A);
  exp2 = RANDOM(x,y);
  exp1 = (exp1 | exp2) + C;
  exp2 = (y-x+1) + x;
  retval = exp1 % exp2;
  return(retval);
} /* end of NURand */


/*****************************************************************/
/*   C r e a t e C L a s t                                       */
/*      Build cust last name from predefined syllables.          */
/*                                                               */
/*  PARMS:                                                       */
/*     iNum => num used to build last name                       */
/*     NameStr => address to store last name                     */
/*****************************************************************/
void CreateCLast(iNum, NameStr)
  int iNum;
  char *NameStr;
{
  static char *sylble[] =
                {"BAR", "OUGHT", "ABLE", "PRE", "PRES",
                 "ESE", "ANTI", "CALLY", "ATION", "EING" };
  strcpy(NameStr, sylble[iNum/100]);
  strcat(NameStr, sylble[(iNum/10)%10]);
  strcat(NameStr, sylble[(iNum%10)]);

  return;
} /* end of CreateCLast */


/*****************************************************************/
/*   C r e a t e S t r                                           */
/*      Create string of random alphanumeric characters.         */
/*                                                               */
/*  PARMS:                                                       */
/*     iMin => minimum length of string                          */
/*     iMax => maximum length of string                          */
/*     szTemp => address to store created string                 */
/*  If string is not variable length, then set iMin and iMax to  */
/*  the actual fixed length.                                     */
/*****************************************************************/
void CreateStr(iMin, iMax, szTemp)
  int iMin, iMax;
  char *szTemp;
{
  int iAlphalen,
      i, j;

  iAlphalen = RANDOM(iMin,iMax); /* determine str len  */
  memset(szTemp, '\0', strlen(szTemp));
  for (j=0; j<iAlphalen; j++)
  {
```

```
    i = RANDOM(0,61);
    szTemp[j]= chAlpha[i];
  }
  return;
} /* end of CreateStr */


/*********************************************************************/
/*   G e t D a t e T i m e                                           */
/*      Get current date/time stamp.                                 */
/*                                                                   */
/*  PARMS:                                                           */
/*     szTime => pointer to time string                              */
/*     szDate => pointer to date string                              */
/*********************************************************************/
void GetDateTime(szTime, szDate)
  char *szTime;
  char *szDate;
{
  char tstr[5];
  static struct tm *DateTime;
  short iYear;

  time(&ltime);   /* get current date & time */
  DateTime = gmtime(&ltime);
    /* return time */
  sprintf(tstr, "%02d", DateTime->tm_hour);
  strcpy(szTime, tstr);
  sprintf(tstr, "%02d", DateTime->tm_min);
  strcat(szTime, tstr);
  sprintf(tstr, "%02d", DateTime->tm_sec);
  strcat(szTime, tstr);
    /* return date */
  sprintf(tstr, "%02d", (DateTime->tm_mon + 1));
  strcpy(szDate, tstr);
  sprintf(tstr, "%02d", DateTime->tm_mday);
  strcat(szDate, tstr);
  iYear = DateTime->tm_year + 1900;
  sprintf(tstr, "%04d", iYear);
  strcat(szDate, tstr);
} /* end of GetDateTime */


/*********************************************************************/
/*   C r e a t e Z i p                                               */
/*      Create zip code string.                                      */
/*                                                                   */
/*  PARMS:                                                           */
/*    Returns zip code string.                                       */
/*********************************************************************/
void CreateZip(zipstr)
char *zipstr;
{
/* Redefine max generated random number. Default is 32,767. */
  int iZ;

  iZ = MaxRand();
  sprintf(zipstr, "%05d", iZ);
  strcat(zipstr, "1111");
  return;
} /* end of CreateZip */


/* Redefine max generated random number. Default is 32,767. */
#undef RAND_MAX
#define RAND_MAX +99999
/*********************************************************************/
/*   M a x R a n d                                                   */
/*      Create random number larger than 32767.                      */
/*  NOTE: This is the rand() function used by AS/400 with one        */
/*      slight modification.  In the rand() function, rand_next is   */
/*      shifted right by 16 rather than the 14 used here.  Shifting  */
/*      by 16 produces a max value of 65535.                         */
/*                                                                   */
/*  PARMS:                                                           */
/*    Returns zip code string.                                       */
/*********************************************************************/
int MaxRand(void)
{
  int rand_temp;

  rand_next = rand_next * 1103515245 + 12345;
  rand_temp = (rand_next >> 14)%((unsigned long)(RAND_MAX)+1);
  return(rand_temp);
} /* end of CreateZip */
```

LOADORD: Fill ORDERS, ORDLIN, and NEWORD Files

```
/*********************************************************************/
/* This program fills the neword, orders, and ordlin */
/* files.  It fills them all simultaneously. It      */
/* will create an order record, selecting the customer for that  */
/* order at random (each customer will have one and only one order  */
/* but which order that will be is determined with random numbers). */
/* The next thing done is only done for orders between 2,101 and     */
/* 3000 for each district, this is the creation of a neworder       */
/* record. The orderline records for that order are created last,   */
/* the number of lines for the  order is between 5 and 15, selected */
/* at random.                                                       */
/*   C  H  A  N  G  E  S     &      U  P  D  A  T  E  S             */
/*                                                                  */
/*                                                                  */
/*                                                                  */
/*    PROGRAM-ID. LOADORD.                                          */
/*    AUTHOR. TPCC.                                                 */
/*    INSTALLATION. ROCHESTER.                                      */
/*    DATE-WRITTEN. 03/07/96.                                       */
/*    DATE-COMPILED. 03/07/96.                                      */
/*********************************************************************/

/* Start include files */
#include <stdlib.h>
#include <errno.h>
#include <recio.h>
#include <string.h>
#include <xxfdbk.h>
#include <milib.h>
#include <mlcomput.h>
#include <decimal.h>
#include <time.h>
#include <xxcvt.h>
 /* */
/* MAPINC generates structures for the database files          */
#pragma mapinc("newordpf","*libl/newordpf(*all)","both key","d _P",,"tpcc")
#include "newordpf"
#pragma mapinc("orderspf","*libl/orderspf(*all)","both key", "d _P",,"tpcc ")
#include "orderspf"
#pragma mapinc("ordlinpf","*libl/ordlinpf(*all)","both key","d _P",,"tpcc" )
#include "ordlinpf"
  /*
        */
        /*     Orders per district     */
#define ORDERSPERDIST 3000
#define MAX_RAND 100000
#define RANDOM(a,b) (rand() % (b-a+1))+a
#define MAXNUM1 600

 void LoadOrders(int, int, int);
 void LoadNeword(int, int, int);
 int GetCstmrID(int);
 int GetItemID(void);
 void GetDateTime(char *, char *);
 void GetCarrier(char *);
 void GetAmount(char *);
 int MaxRand(void);
 void BldArray(void);

 int v;

    _RFILE
          *NewOrder_File,
          *Orders_File,
          *OrderLine_File;

tpcc_ORRCD_both_t OrderRec;
tpcc_OLRCD_both_t OrdlnRec;
tpcc_NORCD_both_t NewordRec;

 static time_t ltime;
 static char TimeStr[7];
 static char DateStr[9];
 static char CarrStr[3];
 static char AmntStr[8];
 static int ocid;
 static unsigned long int rand_next = 1;
 int  numarray[ORDERSPERDIST+1];
  /*
        */

        /*    There are 10 entries of length 24 each  */
  char  *DIST_INFOâã =
      {"Info for District Num 00",
       "Info for District Num 01",
       "Info for District Num 02",
       "Info for District Num 03",
       "Info for District Num 04",
       "Info for District Num 05",
       "Info for District Num 06",
       "Info for District Num 07",
       "Info for District Num 08",
       "Info for District Num 09",
       "Info for District Num 10" };
    /*
            */
  /* There are 3 sets of characters that are concatenated */
  /* to make the district information fields for Stock */
char *DistInf1[] = {
  "abcdefghijk", "nowisthetim", "thisiswhere",
  "whatdoyouno", "notmuchthen", "whydoyouask",
  "Idonotknow_", "sowhereisit", "inthecities",
  "andthetowns"};

char *DistInf2[] = {
  "01", "02", "03", "04", "05", "06", "07", "08", "09", "10"};

char *DistInf3[] = {
  "mnopqrstuvw", "whydidyoudo", "interactive",
  "batchwork__", "systemview_", "performance",
  "Ineverwork_", "Ialwaysplay", "nexttolast_",
  "nowthefinal"};

    /*
            */
            char zeroesd[8] = "00000000";
            char zeroest[6] = "000000";
    /*
            */
            char *CarId[] = { "00","01", "02", "03", "04", "05",
                              "06", "07", "08", "09", "10" };
    /*
            */

int main(argc, argv)
register int argc; char **argv;
{
   short int i;
   short int iStrWrhs=1;
   short int iEndWrhs=1;
   short int iCurWrhs=1;
   short int iCurDist;
   int iCurOid;
   int rcode = 0;


  if( (Orders_File = _Ropen("orderspf","rr+","blkrcd=Y")) == NULL)
  {
    printf("Open of orders file failed\n");
    exit(99);
  }
  if( (OrderLine_File = _Ropen("ordlinpf","rr+","blkrcd=Y")) == NULL)
  {
    printf("Open of Orderline file failed\n");
    exit(99);
  }
  if( (NewOrder_File = _Ropen("newordpf","rr+","blkrcd=Y")) == NULL)
  {
    printf("Open of NewOrder file failed\n");
    exit(99);
  }

    /* Set up random number generator */
  time(&ltime);
  srand(ltime);          /* seed the random num generator */
  rand_next = (unsigned long int)ltime;

  BldArray();
    /* parse the input parms */
  for (argc--, argv++, i=0; argc>0; ++i, argc--, argv++)
  {
    if (i==0) {
      iStrWrhs = atoi(*argv);
    }
    else  {
      iEndWrhs = atoi(*argv);
    }
  }  /* end of parsing input parms */

  for (iCurOid=1; iCurOid<=ORDERSPERDIST; iCurOid++)
  {
```

```
    for (iCurWrhs=iStrWrhs; iCurWrhs<=iEndWrhs; iCurWrhs++)
    {

      for (iCurDist=1; iCurDist<=10; iCurDist++)
      {

        LoadOrders(iCurWrhs, iCurDist, iCurOid);
        if(iCurOid >= 2101)
        {
          LoadNeword(iCurWrhs, iCurDist, iCurOid);
        }
      }
    }
   }
 _Rclose(Orders_File);
 _Rclose(OrderLine_File);
 _Rclose(NewOrder_File);

 return(rcode);
} /* end of main */


/*********************************************************************/
/*   L o a d O r d e r s                                             */
/*      Create & write orders and Orderline rec.                     */
/*                                                                   */
/*  PARMS:                                                           */
/*     iCurWrhs  => current warehouse id                             */
/*     iCurDist  => current district id                              */
/*     iCurOid   => current Order id                                 */
/*********************************************************************/
void LoadOrders(iCurWrhs, iCurDist, iCurOid)
  short int iCurWrhs;
  short int iCurDist;
  int iCurOid;
{
  decimal(7,2) d=0.00;
  int i,tmpvar;
  short int oln,curoln;
  char StockDist[25];

  OrderRec.OID  = OrdlnRec.OLOID = iCurOid;
  OrderRec.OWID = OrdlnRec.OLWID = iCurWrhs;
  OrderRec.ODID = OrdlnRec.OLDID = iCurDist;

     OrderRec.OCID = GetCstmrID(iCurOid);
     GetDateTime(TimeStr, DateStr);
     strncpy(OrderRec.OENTDT,DateStr,strlen(DateStr));
     strncpy(OrderRec.OENTTM,TimeStr,strlen(TimeStr));
     if(iCurOid<2101)
     {
       GetCarrier(CarrStr);
       strncpy(OrderRec.OCARID,CarrStr,strlen(CarrStr));
     }
     else
     {
       memset(OrderRec.OCARID,'\0', strlen(OrderRec.OCARID));
     }
     OrderRec.OLINES = RANDOM(5, 15);
     curoln = OrderRec.OLINES;
     OrderRec.OLOCAL = 1;
     _Rwrite(Orders_File,(void *)&OrderRec, sizeof(OrderRec));
     for(oln=1;oln<=curoln;oln++)
     {
         OrdlnRec.OLOID = iCurOid;
         OrdlnRec.OLNBR = oln;
         while((OrdlnRec.OLIID = MaxRand())==0);
         OrdlnRec.OLSPWH = iCurWrhs;
         OrdlnRec.OLQTY = 5;
         if(iCurOid<2101)
         {
strncpy(OrdlnRec.OLDLVD, OrderRec.OENTDT, sizeof(OrderRec.OENTDT));
strncpy(OrdlnRec.OLDLVT, OrderRec.OENTTM, sizeof(OrderRec.OENTTM));
           OrdlnRec.OLAMNT = 0.00;
         }
         else
         {
           strncpy(OrdlnRec.OLDLVD, zeroesd, 8);
           strncpy(OrdlnRec.OLDLVT, zeroest, 6);
           tmpvar = MaxRand();
           d = tmpvar / 100;
           OrdlnRec.OLAMNT = d;
         }
     tmpvar = RANDOM(0, 9);
    memcpy(StockDist, DistInf1[tmpvar], 11);
    memcpy((void *)&StockDist[11], DistInf2[iCurDist-1],2);
     tmpvar = RANDOM(0, 9);
    memcpy((void *)&StockDist[13], DistInf3[tmpvar],11);
    strncpy(OrdlnRec.OLDSTI, StockDist, 24);
         _Rwrite(OrderLine_File,(void *)&OrdlnRec, sizeof(OrdlnRec));
     } /* End of order line loop  */
 return;
}      /*End of LoadOrders  */



/*********************************************************************/
/*   L o a d N e w o r d                                             */
/*      Create & write neword rec.                                   */
/*                                                                   */
/*  PARMS:                                                           */
/*     iCurWrhs  => current warehouse id                             */
/*     iCurDist  => current district id                              */
/*     iCurOid   => current order id                                 */
/*********************************************************************/
void LoadNeword(iCurWrhs, iCurDist, iCurOid)
  short int iCurWrhs;
  short int iCurDist;
  short int iCurOid;
{

  NewordRec.NOWID = iCurWrhs;
  NewordRec.NODID = iCurDist;
  NewordRec.NOOID = iCurOid;
  _Rwrite(NewOrder_File,(void *)&NewordRec, sizeof(NewordRec));
  return;
}      /*End of LoadNeword  */

/*********************************************************************/
/*   G e t C s t m r I D                                             */
/*      Generates a random customer id.                              */
/*                                                                   */
/*  PARMS:                                                           */
/*     oid => Order ID                                               */
/*********************************************************************/
int GetCstmrID(oid)
int oid;
{


  if( (OrderRec.OWID == 1) && (OrderRec.ODID == 1) )
  {
       if (oid==1)
       {         v=RANDOM(1, ORDERSPERDIST);
                 ocid = numarray[v];
       }
      else
       {         v++;
                 if(v > ORDERSPERDIST)
                 {
                 v = 1;
                 }
                 ocid= numarray[v];
       }
  }
  else
  {
    ocid++;
    if(ocid > ORDERSPERDIST)
    {
      ocid = 1;
    }
  }
  return(ocid);
} /* End of GetCstmrID */

/*********************************************************************/
/*   G e t D a t e T i m e                                           */
/*      Get current date/time stamp.                                 */
/*                                                                   */
/*  PARMS:                                                           */
/*     szTime => pointer to time string                              */
/*     szDate => pointer to date string                              */
/*********************************************************************/
void GetDateTime(szTime, szDate)
  char *szTime;
  char *szDate;
{
  char tstrâ5ã;
  static struct tm *DateTime;
  short iYear;

  time(&ltime);   /* get current date & time */
  DateTime = gmtime(&ltime);
    /* return time */
  sprintf(tstr, "%02d", DateTime->tm_hour);
  strcpy(szTime, tstr);
  sprintf(tstr, "%02d", DateTime->tm_min);
  strcat(szTime, tstr);
  sprintf(tstr, "%02d", DateTime->tm_sec);
  strcat(szTime, tstr);
    /* return date */
  sprintf(tstr, "%02d", (DateTime->tm_mon + 1));
  strcpy(szDate, tstr);
  sprintf(tstr, "%02d", DateTime->tm_mday);
  strcat(szDate, tstr);
  iYear = DateTime->tm_year + 1900;
  sprintf(tstr, "%04d", iYear);
  strcat(szDate, tstr);
  return;
} /* end of GetDateTime */


/*********************************************************************/
/*   G e t C a r r i e r                                             */
/*      Get carrier id if oid < 2101.                                */
/*                                                                   */
/*  PARMS:                                                           */
/*     szCarr => pointer to carrier string                           */
/*********************************************************************/
void GetCarrier(szCarr)
  char *szCarr;
{
  char tstr[2];
  int ri;
  ri = RANDOM(1, 10);
  strncpy(szCarr, CarId[ri],strlen(CarId[0]));
  return;
} /* end of GetCarrier */


/*********************************************************************/
/*   M a x R a n d                                                   */
/*      Get item id between 1 & 100,000                              */
/*                                                                   */
/*  PARMS:                                                           */
/*     None.                                                         */
/*********************************************************************/
int MaxRand()
{
  int rand_temp;

  rand_next = rand_next * 1103515245 + 12345;
  rand_temp = (rand_next >> 14)%((unsigned long)(MAX_RAND)+1);
  return(rand_temp);
}
/*********************************************************************/
/*   Build the array for OCIDs.                                      */
/*      Generates a unique random customer id and builds an array.   */
/*                                                                   */
/*  PARMS:                                                           */
/*     None                                                          */
/*********************************************************************/
void BldArray()
{
int i,j,k=0,n=0,x,num,same,y1=0,y2=0,y3=0,y4=0,y5=0,count=0;
int numarray1[MAXNUM1],
    numarray2[MAXNUM1],
    numarray3[MAXNUM1],
    numarray4[MAXNUM1],
    numarray5[MAXNUM1];



    for (j=0; j <= ORDERSPERDIST; j++)
      {         numarray[j] = 0;
                numarray1[j] = 0;
                numarray2[j] = 0;
                numarray3[j] = 0;
                numarray4[j] = 0;
                numarray5[j] = 0;
     }
         x=MAXNUM1;

     for (i=0; i <= ORDERSPERDIST; i++)
       {
          num = RANDOM(1,ORDERSPERDIST);
          same = 0;
        for (j=0; j < x; j++)
          {
           if (num == numarray1[j])   same=1;
           if (num == numarray2[j])   same=1;
           if (num == numarray3[j])   same=1;
           if (num == numarray4[j])   same=1;
           if (num == numarray5[j])   same=1;
          }
          if (same==0) {
```

```
        count++;
        numarray[n++]=num;
        if ((count>=0) && (count<600))
         numarray1[y1++]=num;
        if ((count>=600) && (count<1200))
          numarray2[y2++]=num;
         if ((count>=1200) && (count<1800))
          numarray3[y3++]=num;
         if ((count>=1800) && (count<2400))
          numarray4[y4++]=num;
         if ((count>=2400) && (count<3000))
          numarray5[y5++]=num;
          }
          else i--;
      }
} /* End of BldArray */
```

SETUP: System Setup Program

```
            PGM        PARM(&OBJLIB &BLDLIB &DATLIB &CRTSAVFILE +
                         &SAVLIB &JRNLIB &JRNASP &STRWHC &ENDWHC)

/* This program was changed to submit several stages of it to    */
/* batch. 12/19/93                                               */
/* FURTHER CHANGES MADE 1/07/94                                  */
/* This program is one of the last called during the creation of */
/* the TPCC data base.  It creates the data queue from which     */
/* each user will be assigned a warehouse/district when they     */
/* sign on.  It also creates the job description and user        */
/* profile for the interactive users, next it creates the job    */
/* and subsystem descriptions for the batch jobs which will      */
/* execute the deffered portion of the delivery transaction.     */
/* It then creates the journal and journal receiver which are    */
/* used when a transaction is to be rolled-back or when lost     */
/* transactions need to be re-applied.  After creating the       */
/* journal it reorganizes the customer file to be sorted by      */
/* customer last name and first name, then the customer logical  */
/* file is created.  After all of the files are created the      */
/* journaling of the physical files is begun.  The last thing    */
/* this program does is to save the DB so we can use the restore */
/* function to get the DB back to its initial state.             */

            DCL        VAR(&OBJLIB) TYPE(*CHAR) LEN(10)
            DCL        VAR(&BLDLIB) TYPE(*CHAR) LEN(10)
            DCL        VAR(&DATLIB) TYPE(*CHAR) LEN(10)
            DCL        VAR(&SAVLIB) TYPE(*CHAR) LEN(10)
            DCL        VAR(&JRNLIB) TYPE(*CHAR) LEN(10)
            DCL        VAR(&JRNASP) TYPE(*CHAR) LEN(1)
            DCL        VAR(&CRTSAVFILE) TYPE(*CHAR) LEN(1)
            DCL        VAR(&STRWH) TYPE(*DEC) LEN(4 0)
            DCL        VAR(&ENDWH) TYPE(*DEC) LEN(4 0)
            DCL        VAR(&STRWHC) TYPE(*CHAR) LEN(4)
            DCL        VAR(&ENDWHC) TYPE(*CHAR) LEN(4)
/* Monitor for messages and setup library list                   */

            MONMSG     MSGID(CPF1064) /* Class already exists */
            MONMSG     MSGID(CPD1411) /* Subsystem already exists */
            MONMSG     MSGID(CPD1416) /* Subsystem already active */
            MONMSG     MSGID(CPD1475) /* Routing Entry already +
                         exists */
            MONMSG     MSGID(CPD1535) /* Job Queue Entry already +
                         exists */
            MONMSG     MSGID(CPD1611) /* JOBD already exists */
            MONMSG     MSGID(CPD1621) /* JOBQ not FOUND     */
            MONMSG     MSGID(CPF1621) /* JOBD not created    */
            MONMSG     MSGID(CPF1696) /* Subsystem not created */
            MONMSG     MSGID(CPF1697) /* Subsystem not changed */
            MONMSG     MSGID(CPF2103) /* Library already in list */
            MONMSG     MSGID(CPF2105) /* Object not found       */
            MONMSG     MSGID(CPF2110) /* Object not found       */
            MONMSG     MSGID(CPF2111) /* Library already exists */
            MONMSG     MSGID(CPF2204) /* User Profile Does Not +
                         Exist */
            MONMSG     MSGID(CPF2555) /* Reply list entry exists */
            MONMSG     MSGID(CPF3323) /* Job Queue already exists */
            MONMSG     MSGID(CPF3770) /* No objects saved or    +
                              restored  */
            MONMSG     MSGID(CPF5813) /* File already exists    */
            MONMSG     MSGID(CPF7010) /* Journal already exists */
            MONMSG     MSGID(CPF7032) /* File not journaled     */
            MONMSG     MSGID(CPF7302) /* File not created       */
            MONMSG     MSGID(CPF7311) /* Errors not allowed in  +
                                 DDS                           */
            MONMSG     MSGID(CPF9812) /* File not found         */

            ADDRPYLE   SEQNBR(3103) MSGID(CPA4067) RPY(G) /* +
                         Save of Data over existing Save Files */

            ADDLIBLE   LIB(&BLDLIB)
            ADDLIBLE   LIB(&DATLIB)
            CHGVAR     VAR(&STRWH) VALUE(&STRWHC)
            CHGVAR     VAR(&ENDWH) VALUE(&ENDWHC)



            SNDPGMMSG  MSGID(CPF9898) MSGF(QCPFMSG) +
                         MSGDTA('Creating Logical Files') +
                         TOPGMQ(*EXT) MSGTYPE(*STATUS)
/*  BUILD INDEXES                                                */

            SBMJOB     CMD(CALL PGM(&OBJLIB/ORDERSVIEW) +
                         PARM(&DATLIB &BLDLIB &STRWHC &ENDWHC)) +
                         JOB(ORDERSVIEW) JOBQ(&OBJLIB/TPCBLDJOBQ) +
                         MSGQ(&BLDLIB/TPCCBLDMSG)

            SBMJOB     CMD(CALL PGM(&OBJLIB/ORDLINVIEW) +
                         PARM(&DATLIB &BLDLIB)) JOB(ORDLINVIEW) +
                         JOBQ(&OBJLIB/TPCBLDJOBQ) +
                         MSGQ(&BLDLIB/TPCCBLDMSG)

            SBMJOB     CMD(CALL PGM(&OBJLIB/NEWORDVIEW) +
                          PARM(&DATLIB &BLDLIB)) JOB(NEWORDVIEW) +
                         JOBQ(&OBJLIB/TPCBLDJOBQ) +
                         MSGQ(&BLDLIB/TPCCBLDMSG)

            SBMJOB     CMD(CALL PGM(&OBJLIB/CSTMRVIEW) PARM(&DATLIB +
                         &BLDLIB)) JOB(CSTMRVIEW) +
                         JOBQ(&OBJLIB/TPCBLDJOBQ) +
                         MSGQ(&BLDLIB/TPCCBLDMSG)



/* Creation of TPCC User Profile                                 */
```

```
 ENDIT:     CRTJOBD    JOBD(&OBJLIB/TPCCJOBD) JOBQ(QGPL/QBATCH) +
                         TEXT('Job description for TPCCUSER') +
                         LOG(0) INQMSGRPY(*DFT)
            DLTUSRPRF  USRPRF(TPCCUSER)
            MONMSG     MSGID(CPF2215) EXEC(GOTO CMDLBL(CHGPRF))
            CRTUSRPRF  USRPRF(TPCCUSER) CURLIB(&OBJLIB) +
                         INLPGM(QCMD) TEXT('TPCC User created by +
                         BLDTPCC') SPCAUT(*SAVSYS *ALLOBJ *JOBCTL) +
                         JOBD(&OBJLIB/TPCCJOBD) DLVRY(*HOLD)
            GOTO       CMDLBL(CRTJOBQ)

/* Change the profile if it existed previously                   */

 CHGPRF:    CHGUSRPRF  USRPRF(TPCCUSER) CURLIB(&OBJLIB) +
                         INLPGM(*LIBL/QCMD) TEXT('TPCC User - +
                         Changed by BLDTPCC') SPCAUT(*SAVSYS +
                         *ALLOBJ *JOBCTL) JOBD(&OBJLIB/TPCCJOBD) +
                         DLVRY(*HOLD)


/* Create job queue, class and subsystem description for the   */
/* batch portion of the delivery transaction                   */

 CRTJOBQ:   CRTJOBQ    JOBQ(&OBJLIB/DLVRYJOBQ) TEXT('Delivery Job +
                         Queue')
            CRTCLS     CLS(&OBJLIB/DLVRYCLS) RUNPTY(40) +
                         PURGE(*NO) TEXT('Delivery Job Class')
            CRTSBSD    SBSD(&OBJLIB/BATCHSBS) POOLS((1 *BASE)) +
                         SGNDSPF(QSYS/QDSIGNON)
            ADDJOBQE   SBSD(&OBJLIB/BATCHSBS) +
                         JOBQ(&OBJLIB/DLVRYJOBQ) MAXACT(50) +
                         SEQNBR(1000)
            ADDRTGE    SBSD(&OBJLIB/BATCHSBS) SEQNBR(1000) +
                         CMPVAL(DLVRYJOB) PGM(QSYS/QCMD) +
                         CLS(&OBJLIB/DLVRYCLS)

/* Create the journal receiver in the mirrored user ASP        */

            IF         COND(&STRWHC *EQ '0001') THEN(DO)

            CRTLIB     LIB(&JRNLIB) ASP(&JRNASP)

            CRTMSGQ    MSGQ(&DATLIB/TPCCJRNMSG) TEXT('TPC-C Journal +
                         Hit Threshold Message')
            MONMSG     MSGID(CPF2112)

            CRTJRNRCV  JRNRCV(&JRNLIB/TPCCJRNRCV)              +
                         THRESHOLD(1500000) TEXT('TPCC Journal +
                         Receiver')
            CRTJRNRCV  JRNRCV(&JRNLIB/TEMPRCV)                 +
                         TEXT('TPCC Temporary Receiver (only used +
                         in cleanup pgm)')
            CRTJRN     JRN(&DATLIB/TPCCJRN) +
                         JRNRCV(&JRNLIB/TPCCJRNRCV) +
                         MSGQ(&DATLIB/TPCCJRNMSG) +
                         MNGRCV(*SYSTEM) RCVSIZOPT(*MINFIXLEN) +
                         TEXT('TPCC Journal')
            ENDDO

            CHGJRN     JRN(&DATLIB/TPCCJRN) JRNRCV(*GEN) +
                         MSGQ(&DATLIB/TPCCJRNMSG) MNGRCV(*SYSTEM) +
                         RCVSIZOPT(*MINFIXLEN)

/* WAIT FOR COMPLETION OF REORGANIZES                            */

            RCVMSG     MSGQ(&BLDLIB/TPCCBLDMSG) MSGTYPE(*ANY) +
                         WAIT(*MAX)
            RCVMSG     MSGQ(&BLDLIB/TPCCBLDMSG) MSGTYPE(*ANY) +
                         WAIT(*MAX)
            RCVMSG     MSGQ(&BLDLIB/TPCCBLDMSG) MSGTYPE(*ANY) +
                         WAIT(*MAX)
            RCVMSG     MSGQ(&BLDLIB/TPCCBLDMSG) MSGTYPE(*ANY) +
                         WAIT(*MAX)


/* Start journaling                                              */

            CALL       PGM(&BLDLIB/STRJRNTPCC)

/* Save of TPCC Data Files to Save Files.  Used in Cleanup Program */

            IF         COND(&CRTSAVFILE = 'Y') THEN(DO)
              CRTLIB     LIB(&SAVLIB) TEXT(&DATLIB *CAT ' SAVF')
              CLRLIB     LIB(&SAVLIB)
              CRTSAVF    FILE(&SAVLIB/&DATLIB) TEXT(&DATLIB *CAT ' +
                         SAVF')
              SAVLIB     LIB(&DATLIB) DEV(*SAVF) +
                         SAVF(&SAVLIB/&DATLIB) ACCPTH(*YES)
              MONMSG CPF3701 /*IGNORE WHEN CANNOT SAVE MSGQ*/
            ENDDO

/* Submit this job AFTER save is done or save will be incomplete  */

 ENDPGM:    ENDPGM
```

ORDERSVIEW: Creates Orders View Logical Files

```
            PGM        PARM(&DATALIB &SRCLIB &STRWHC &ENDWHC)
            DCL        VAR(&STRWH) TYPE(*DEC) LEN(4 0)
            DCL        VAR(&ENDWH) TYPE(*DEC) LEN(4 0)
            DCL        VAR(&STRWHC) TYPE(*CHAR) LEN(4)
            DCL        VAR(&ENDWHC) TYPE(*CHAR) LEN(4)
            DCL        VAR(&DATALIB) TYPE(*CHAR) LEN(10)
            DCL        VAR(&PARMSTR) TYPE(*CHAR) LEN(18)
            DCL        VAR(&SRCLIB) TYPE(*CHAR) LEN(10)
            /* CREATE THE UNIQUE VIEW OVER ORDERSPF */
            CHKOBJ     OBJ(&DATALIB/ORDERSLF) OBJTYPE(*FILE)
            MONMSG     MSGID(CPF9801) EXEC(CRTLF +
                         FILE(&DATALIB/ORDERSLF) +
                         SRCFILE(&SRCLIB/QDDSSRC) LVLCHK(*NO))
            /* FOR AUDIT-READY DATA BASES, GENERATE CORRECT +
               CUSTOMER ID'S FOR PRE-DEFINED ORDERS */
            CHKOBJ     OBJ(&DATALIB/ORDERS) OBJTYPE(*FILE)
            MONMSG     MSGID(CPF9801) EXEC(CRTLF +
                         FILE(&DATALIB/ORDERS) +
                         SRCFILE(&SRCLIB/QDDSSRC) LVLCHK(*NO))

            ENDPGM
```

ORDLINVIEW: Creates Orders View Logical Files

```
PGM         PARM(&DATALIB &SRCLIB)
DCL         VAR(&DATALIB) TYPE(*CHAR) LEN(10)
DCL         VAR(&SRCLIB) TYPE(*CHAR) LEN(10)
/* CREATE THE UNIQUE VIEW OVER ORDLINPF */
CHKOBJ      OBJ(&DATALIB/ORDLIN) OBJTYPE(*FILE)
MONMSG      MSGID(CPF9801) EXEC(CRTLF +
              FILE(&DATALIB/ORDLIN) +
              SRCFILE(&SRCLIB/QDDSSRC) LVLCHK(*NO))
/* CREATE PARTIAL KEY VIEW */
CHKOBJ      OBJ(&DATALIB/ORDLINLF) OBJTYPE(*FILE)
MONMSG      MSGID(CPF9801) EXEC(CRTLF +
              FILE(&DATALIB/ORDLINLF) +
              SRCFILE(&SRCLIB/QDDSSRC) LVLCHK(*NO))
ENDPGM
```

NEWORDVIEW: Creates New Orders View Logical Files

```
PGM         PARM(&DATALIB &SRCLIB)
DCL         VAR(&DATALIB) TYPE(*CHAR) LEN(10)
DCL         VAR(&SRCLIB) TYPE(*CHAR) LEN(10)
/* CREATE THE UNIQUE VIEW OVER NEWORDPF */
CHKOBJ      OBJ(&DATALIB/NEWORD) OBJTYPE(*FILE)
MONMSG      MSGID(CPF9801) EXEC(CRTLF +
              FILE(&DATALIB/NEWORD) +
              SRCFILE(&SRCLIB/QDDSSRC) LVLCHK(*NO))
/* CREATE PARTIAL KEY VIEW */
CHKOBJ      OBJ(&DATALIB/NEWORDLF) OBJTYPE(*FILE)
MONMSG      MSGID(CPF9801) EXEC(CRTLF +
              FILE(&DATALIB/NEWORDLF) +
              SRCFILE(&SRCLIB/QDDSSRC) LVLCHK(*NO))
ENDPGM
```

CSTMRVIEW: Creates Customer View Logical Files

```
PGM         PARM(&DATALIB &SRCLIB)
DCL         VAR(&DATALIB) TYPE(*CHAR) LEN(10)
DCL         VAR(&SRCLIB) TYPE(*CHAR) LEN(10)
/* CREATE THE LAST/FIRST NAME VIEW OVER CSTMRPF*/
CHKOBJ      OBJ(&DATALIB/CSTMRLFCRT) OBJTYPE(*FILE)
MONMSG      MSGID(CPF9801) EXEC(CRTLF +
              FILE(&DATALIB/CSTMRLFCRT) +
              SRCFILE(&SRCLIB/QDDSSRC) LVLCHK(*NO))
/* CREATE THE VIEW FOR ACCESS BY LAST NAME */
CHKOBJ      OBJ(&DATALIB/CSTMRLFNAM) OBJTYPE(*FILE)
MONMSG CPF9801 EXEC( +
CRTLF       FILE(&DATALIB/CSTMRLFNAM)  +
              SRCFILE(&SRCLIB/QDDSSRC) LVLCHK(*NO))
ENDPGM
```

STRJRNTPCC: Start Journaling for all TPCC Physical Files

```
      PGM

/************************************************************/
/*                                                          */
/* FUNCTION: THIS PROGRAM STRARTS JOURNALING FOR ALL        */
/*           PHYSICAL FILES FOR THE TPCC WORKLOAD.          */
/*                                                          */
/* INVOCATION:  THIS PROGRAM IS CALLED VIA:                 */
/*                                                          */
/*           CALL  PGM(STRJRNTPCC)                          */
/*           CALLED FROM SETUP AND CLEANUP                  */
/*                                                          */
/* INPUT:   NONE                                            */
/*                                                          */
/* OUTPUT:  JOURNALING START FOR ALL TPCC FILES             */
/*                                                          */
/* DEPENDENCIES AND ASSUMPTIONS:   NONE                     */
/*                                                          */
/************************************************************/

/* Monitor for messages                                     */

            MONMSG      MSGID(CPF7032)
            MONMSG      MSGID(CPF7030)

/* Start journaling on the physical files */

            STRJRNPF    FILE(CSTMRPF) +
                          JRN(TPCCJRN) IMAGES(*BOTH) +
                          OMTJRNE(*OPNCLO)
            STRJRNPF    FILE(DSTRCT) JRN(TPCCJRN) +
                          IMAGES(*BOTH) +
                          OMTJRNE(*OPNCLO)
            STRJRNPF    FILE(HSTRY) +
                          JRN(TPCCJRN) IMAGES(*BOTH) +
                          OMTJRNE(*OPNCLO)
            STRJRNPF    FILE(ITEM) JRN(TPCCJRN) +
                          IMAGES(*BOTH) +
                          OMTJRNE(*OPNCLO)
            STRJRNPF    FILE(NEWORDPF) JRN(TPCCJRN) +
                          IMAGES(*BOTH) +
                          OMTJRNE(*OPNCLO)
            STRJRNPF    FILE(ORDERSPF) JRN(TPCCJRN) +
                          IMAGES(*BOTH) +
                          OMTJRNE(*OPNCLO)
            STRJRNPF    FILE(STOCKPF) JRN(TPCCJRN) +
                          IMAGES(*BOTH) +
                          OMTJRNE(*OPNCLO)
            STRJRNPF    FILE(WRHS) JRN(TPCCJRN) +
                          IMAGES(*BOTH) +
                          OMTJRNE(*OPNCLO)
            STRJRNPF    FILE(ORDLINPF) +
                          JRN(TPCCJRN) IMAGES(*BOTH) +
                          OMTJRNE(*OPNCLO)

ENDPGM:     ENDPGM
```

CRTENVPGMS: Create Environment Programs

```
            PGM         PARM(&APPLIB &DATLIB)
            DCL         VAR(&APPLIB) TYPE(*CHAR) LEN(10)
            DCL         VAR(&DATLIB) TYPE(*CHAR) LEN(10)

/* This CL program compiles all of the CL, C and CBL programs   */
/* that are needed to do a complete rebuild of the database.    */
/* They are compiled in the order they are needed. None of the  */
/* programs are called from this program, only compiled.        */

            /* ADD CODE TO BUILD APPP PROGRAMS */
            MONMSG CPF0000
            ADDLIBLE    LIB(&DATLIB)
            MONMSG      MSGID(CPF2103)
            ADDLIBLE    LIB(&APPLIB)
            MONMSG      MSGID(CPF2103)
            CRTCLPGM    PGM(&APPLIB/ATPCCMTRA) SRCFILE(&APPLIB/QCLSRC)
            CRTCLPGM    PGM(&APPLIB/ATPCCMTRD) SRCFILE(&APPLIB/QCLSRC)
            CRTCLPGM    PGM(&APPLIB/ATPCCMTRS) SRCFILE(&APPLIB/QCLSRC)
            CRTSQLCBLI OBJ(&APPLIB/NOPAYOSMTR) +
                         SRCFILE(&APPLIB/QSQLCBLSRC) COMMIT(*ALL) +
                         OPTION(*NOGEN)
            CRTCBLMOD   MODULE(&APPLIB/NOPAYOSMTR) +
                         SRCFILE(QTEMP/QSQLTEMP) +
                         OPTION(*NOMONOPRC) DBGVIEW(*SOURCE) +
                         OPTIMIZE(*FULL) LINKLIT(*PRC)
            CRTPGM      PGM(&APPLIB/NOPAYOSMTR) +
                         MODULE(&APPLIB/NOPAYOSMTR) +
                         BNDSRVPGM(QSYS/QDBRUNHA) ACTGRP(*CALLER)
            CRTCBLMOD   MODULE(&APPLIB/STKLVLMTR) +
                         SRCFILE(&APPLIB/QLBLSRC) +
                         OPTION(*NOMONOPRC) DBGVIEW(*SOURCE) +
                         OPTIMIZE(*FULL) LINKLIT(*PRC)
            CRTPGM      PGM(&APPLIB/STKLVLMTR) +
                         MODULE(&APPLIB/STKLVLMTR) +
                         BNDSRVPGM(QSYS/QDBRUNHA) ACTGRP(*CALLER)
            CRTCBLMOD   MODULE(&APPLIB/DLVRYMTR) +
                         SRCFILE(&APPLIB/QLBLSRC) +
                         OPTION(*NOMONOPRC) DBGVIEW(*SOURCE) +
                         OPTIMIZE(*FULL) LINKLIT(*PRC)
            CRTPGM      PGM(&APPLIB/DLVRYMTR) +
                         MODULE(&APPLIB/DLVRYMTR) +
                         BNDSRVPGM(QSYS/QDBRUNHA) ACTGRP(*CALLER)

            CLRLIB QRPLOBJ

            ENDPGM
```

C_CREATE: Create ITEM Hash

```
#include <stdlib.h>
#include <recio.h>
#include <stddef.h>
#include <string.h>
#include <stdio.h>
#include <errno.h>

/*============================================================*/
/* internal defines                                           */
/*============================================================*/

void main(argc, argv)
register int argc; char *argv[];
{
  long xx;
  char flag;
  char hash_name[10];
  char pf[10];
  char pf_lib[10];
  char lf[10];
  char lf_lib[10];
  char expression[255];

  typedef _Packed struct key_struct {
                          char name[10];
                          long value;
     } key_ranges_structure[5];

  key_ranges_structure key_ranges;

  flag = '1';
  memcpy(hash_name ,"ITEMLF    ",10);
  memcpy(pf        ,"ITEM      ",10);
  memcpy(lf        ,"ITEMLF    ",10);
  memcpy(pf_lib    ,"TPCCD698  ",10);
  memcpy(lf_lib    ,"TPCCD698  ",10);
  pf_lib[10] = '\0';
  lf_lib[10] = '\0';
  memcpy(expression,"*DFT                          ",33);
  expression[33] = '\0';

  key_ranges[0].value = 100000;

  xx=qdbcrtha(flag,hash_name,pf,pf_lib,
              lf,lf_lib,key_ranges,expression);
```

```
  xx =0;

}
```

C_CREATE2: Create STOCK Hash

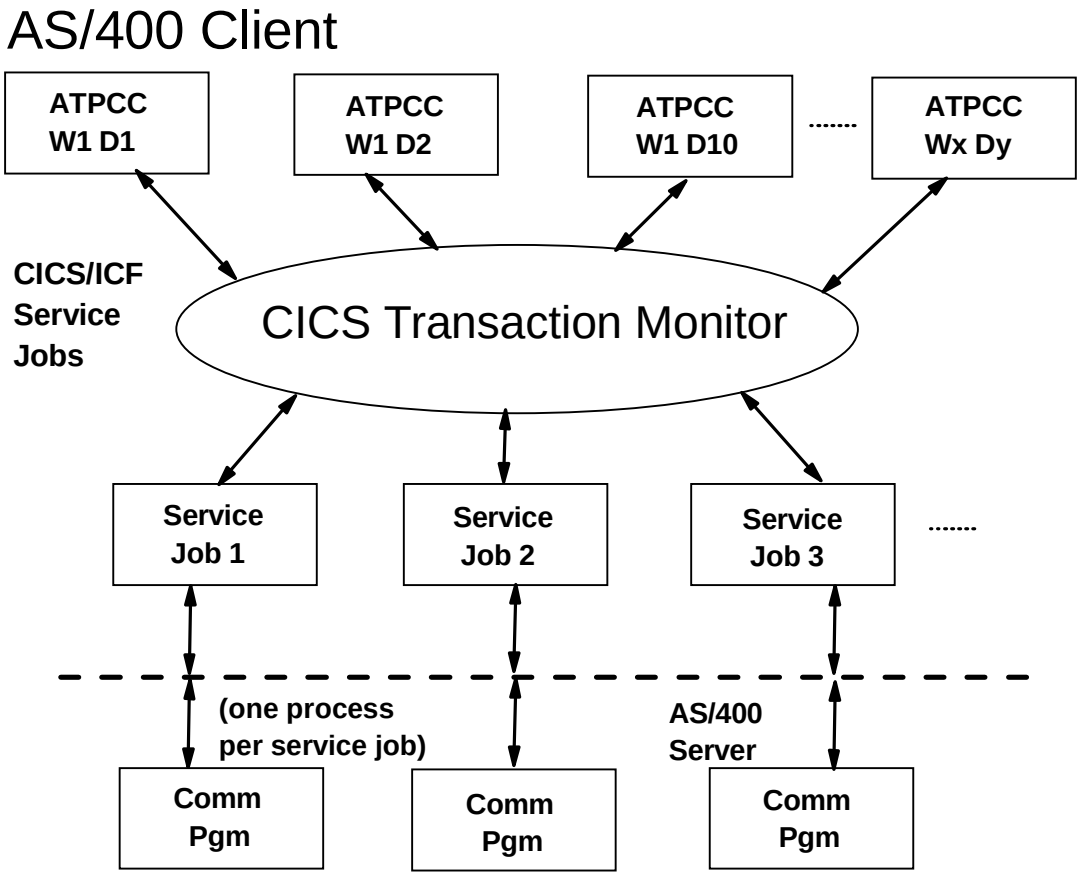
```
#include <stdlib.h>
#include <recio.h>
#include <stddef.h>
#include <string.h>
#include <stdio.h>
#include <errno.h>

/*============================================================*/
/* internal defines                                           */
/*============================================================*/

void main(argc, argv)
register int argc; char *argv[];
{
  long xx;
  char flag;
  char hash_name[10];
  char pf[10];
  char pf_lib[10];
  char lf[10];
  char lf_lib[10];
  char expression[255];

  typedef _Packed struct key_struct {
                       char name[10];
                       long value;
     } key_ranges_structure[5];

  key_ranges_structure key_ranges;

  flag = '0';
  memcpy(hash_name ,"STOCK     ",10);
  memcpy(pf        ,"STOCKPF   ",10);
  memcpy(lf        ,"STOCK     ",10);
  memcpy(pf_lib    ,"TPCCD698  ",10);
  memcpy(lf_lib    ,"TPCCD698  ",10);
  pf_lib[10] = '\0';
  lf_lib[10] = '\0';
  memcpy(expression,"*DFT                          ",33);
  expression[33] = '\0';

  key_ranges[0].value = 3700;
  key_ranges[1].value = 100000;

  xx=qdbcrtha(flag,hash_name,pf,pf_lib,
              lf,lf_lib,key_ranges,expression);

  xx =0;

}
```

C_CREATE3: Create CSTMR Hash

```
#include <stdlib.h>
#include <recio.h>
#include <stddef.h>
#include <string.h>
#include <stdio.h>
#include <errno.h>

/*============================================================*/
/* internal defines                                           */
/*============================================================*/

void main(argc, argv)
register int argc; char *argv[];
{
  long xx;
  char flag;
  char hash_name[10];
  char pf[10];
  char pf_lib[10];
  char lf[10];
  char lf_lib[10];
  char expression[255];

  typedef _Packed struct key_struct {
                       char name[10];
                       long value;
     } key_ranges_structure[5];

  key_ranges_structure key_ranges;

  flag = '0';
  memcpy(hash_name ,"CSTMR     ",10);
  memcpy(pf        ,"CSTMRPF   ",10);
  memcpy(lf        ,"CSTMR     ",10);
  memcpy(pf_lib    ,"TPCCD698  ",10);
  memcpy(lf_lib    ,"TPCCD698  ",10);
  pf_lib[10] = '\0';
  lf_lib[10] = '\0';
  memcpy(expression,"*DFT                          ",33);
  expression[33] = '\0';

  key_ranges[0].value = 3000;
  key_ranges[1].value = 10;
  key_ranges[2].value = 3700;

  xx=qdbcrtha(flag,hash_name,pf,pf_lib,
              lf,lf_lib,key_ranges,expression);

  xx =0;

}
```

Appendix D. Application Source Code

Program Flow.



The main program on the Client System communicates with a set of CICS service jobs on the client. Each service job has a corresponding job on the server. Communication between the CICS service job on the client and the remote job on the server takes the place through an ICF file.

The communication programs on the server handle all five TPCC transactions.

STRICFOLDS: Start CICS Service Jobs

```
#include <stddef.h>
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <aegtminc.h>

/*================================================================*/
/* start transaction monitor                                      */
/*================================================================*/

void main(int argc, char *argv[])
{
   error_structure err;
   short xx;
   char rmtloc[8], mode[8];
   int nojobs, payjobs, stkjobs, dlvjobs, osjobs;

   monitor_type monitor;
   monitor.regions = 1;
   monitor.transactions = 5;

   memcpy(rmtloc, argv[1], 8);
   memcpy(mode,   argv[2], 8);
   nojobs  =  atoi(argv[3]);
   payjobs =  atoi(argv[4]);
   stkjobs =  atoi(argv[5]);
   dlvjobs =  atoi(argv[6]);
   osjobs  =  atoi(argv[7]);

/* Set up parameters for starting the transaction monitor       */

   monitor.tarea[1].ttrans = 66;
   memcpy(monitor.tarea[1].icf_file,"TPCCKVCLNT/STOCICFF  ",21);
   memcpy(monitor.tarea[1].remote_loc,argv[1],8);
   memcpy(monitor.tarea[1].mode,argv[2],8);
   memcpy(monitor.tarea[1].authority[1],"TPCCUSER",8);
   monitor.tarea[1].job_priority = 0;
   monitor.tarea[1].send_length = 230;
   monitor.tarea[1].recv_length = 670;
   monitor.tarea[1].time_out     = 0;

   monitor.tarea[2].ttrans = 99;
   memcpy(monitor.tarea[2].icf_file,"TPCCKVCLNT/DLVRICFF  ",21);
   memcpy(monitor.tarea[2].remote_loc,argv[1],8);
   memcpy(monitor.tarea[2].mode,argv[2],8);
   memcpy(monitor.tarea[2].authority[1],"TPCCUSER",8);
   monitor.tarea[2].job_priority = 0;
   monitor.tarea[2].send_length = 230;
   monitor.tarea[2].recv_length = 670;
   monitor.tarea[2].time_out     = 0;

   monitor.tarea[3].ttrans = 77;
   memcpy(monitor.tarea[3].icf_file,"TPCCKVCLNT/MAINICFF  ",21);
   memcpy(monitor.tarea[3].remote_loc,argv[1],8);
   memcpy(monitor.tarea[3].mode,argv[2],8);
   memcpy(monitor.tarea[3].authority[1],"TPCCUSER",8);
   monitor.tarea[3].job_priority = 0;
   monitor.tarea[3].send_length = 230;
   monitor.tarea[3].recv_length = 670;
   monitor.tarea[3].time_out     = 0;

   monitor.tarea[4].ttrans = 88;
   memcpy(monitor.tarea[4].icf_file,"TPCCKVCLNT/MAINICFF  ",21);
   memcpy(monitor.tarea[4].remote_loc,argv[1],8);
   memcpy(monitor.tarea[4].mode,argv[2],8);
   memcpy(monitor.tarea[4].authority[1],"TPCCUSER",8);
   monitor.tarea[4].job_priority = 0;
   monitor.tarea[4].send_length = 230;
   monitor.tarea[4].recv_length = 670;
   monitor.tarea[4].time_out     = 0;

   monitor.tarea[5].ttrans = 44;
   memcpy(monitor.tarea[5].icf_file,"TPCCKVCLNT/MAINICFF  ",21);
   memcpy(monitor.tarea[5].remote_loc,argv[1],8);
   memcpy(monitor.tarea[5].mode,argv[2],8);
   memcpy(monitor.tarea[5].authority[1],"TPCCUSER",8);
   monitor.tarea[5].job_priority = 0;
   monitor.tarea[5].send_length = 230;
   monitor.tarea[5].recv_length = 670;
   monitor.tarea[5].time_out     = 0;

 /*      The transaction types                                   */
 /*      66  =  Stock Level                                      */
 /*      99  =  Delivery                                         */
 /*      77  =  New Order                                        */
 /*      88  =  Payment                                          */
 /*      44  =  Order Status                                     */

   monitor.rarea[1].ttypes = 5;

   monitor.rarea[1].ttrans[1] = 66;
   monitor.rarea[1].max_jobs[1] = stkjobs;

   monitor.rarea[1].ttrans[2] = 99;
   monitor.rarea[1].max_jobs[2] = dlvjobs;

   monitor.rarea[1].ttrans[3] = 77;
   monitor.rarea[1].max_jobs[3] = nojobs;

   monitor.rarea[1].ttrans[4] = 88;
   monitor.rarea[1].max_jobs[4] = payjobs;

   monitor.rarea[1].ttrans[5] = 44;
   monitor.rarea[1].max_jobs[5] = osjobs;

/* Invoke the start transaction monitor function with the right parms*/

   xx = aegtmstr(monitor,&err);
   xx = xx;
}
```

FIRSTPGM1: Sign-On Program

```
            PGM
            DCL &WRHS *CHAR 4
            DCL &DST *CHAR 2
            DCL         VAR(&QNAME) TYPE(*CHAR) LEN(10) VALUE(INITDTAQ)
            DCL         VAR(&QLIB) TYPE(*CHAR) LEN(10) +
                          VALUE(TPCCKVCLNT)
            DCL         VAR(&QNAME1) TYPE(*CHAR) LEN(10) +
                          VALUE(TEMPDTAQ)
            DCL         VAR(&QLIB1) TYPE(*CHAR) LEN(10) VALUE(QTEMP)
            DCL         VAR(&FLDLEN) TYPE(*DEC) LEN(5 0) VALUE(6)
            DCL         VAR(&WAIT) TYPE(*DEC) LEN(5 0) VALUE(0)
            DCL         VAR(&VALUE) TYPE(*CHAR) LEN(6)
            CALL        PGM(QRCVDTAQ) PARM(&QNAME &QLIB &FLDLEN +
                          &VALUE &WAIT)
            CRTDTAQ     DTAQ(&QLIB1/&QNAME1) MAXLEN(6)
            CALL        PGM(QSNDDTAQ) PARM(&QNAME1 &QLIB1 &FLDLEN +
                          &VALUE)
            CHGVAR      VAR(&WRHS)     VALUE(%SST(&VALUE 1 4))
            CHGVAR      VAR(&DST)   VALUE(%SST(&VALUE 5 2))
            ?           SECONDGM2 WRHSNUM(&WRHS) DSTRCTNUM(&DST)


ENDIT:      ENDPGM
```

FIRSTPGM2: Sign-On Program

```
            PGM PARM(&SSNWHSE &SSNDSTRCT)
            DCL         VAR(&SSNWHSE) TYPE(*CHAR) LEN(4)
            DCL         VAR(&SSNWHSED) TYPE(*DEC) LEN(4 0)
            DCL         VAR(&SSNDSTRCT) TYPE(*CHAR) LEN(2)
            DCL         VAR(&STRWRHS) TYPE(*CHAR) LEN(4)
            DCL         VAR(&NUMUSERS) TYPE(*CHAR) LEN(5)
            DCL         VAR(&STRWRHSD) TYPE(*DEC) LEN(4 0)
            DCL         VAR(&NUMUSERSD) TYPE(*DEC) LEN(5 0)
            DCL         VAR(&LIB) TYPE(*CHAR) LEN(10) VALUE(TPCCKVCLNT)
            RTVDTAARA   DTAARA(&LIB/STRWRHS) RTNVAR(&STRWRHS)
            RTVDTAARA   DTAARA(&LIB/NUMUSERS) RTNVAR(&NUMUSERS)
            CHGVAR &STRWRHSD &STRWRHS
            CHGVAR &NUMUSERSD &NUMUSERS
            CHGVAR &SSNWHSED &SSNWHSE
            SNDPGMMSG &SSNWHSE
            SNDPGMMSG &STRWRHS
            IF          COND(&SSNWHSED *LE (&STRWRHSD + 100)) +
                          THEN(STRCICSUSR CTLRGN(TPCC) TRANID(TPCC))
            IF          COND((&SSNWHSED *GT (&STRWRHSD + 100)) & +
                          (&SSNWHSED *LE (&STRWRHSD + 200))) +
                          THEN(STRCICSUSR CTLRGN(TPC0) TRANID(TPC0))
            IF          COND((&SSNWHSED *GT (&STRWRHSD + 200)) & +
                          (&SSNWHSED *LE (&STRWRHSD + 300))) +
                          THEN(STRCICSUSR CTLRGN(TPC1) TRANID(TPC1))
            IF          COND((&SSNWHSED *GT (&STRWRHSD + 300)) & +
                          (&SSNWHSED *LE (&STRWRHSD + 400))) +
                          THEN(STRCICSUSR CTLRGN(TPC2) TRANID(TPC2))
            IF          COND((&SSNWHSED *GT (&STRWRHSD + 400)) & +
                          (&SSNWHSED *LE (&STRWRHSD + 500))) +
                          THEN(STRCICSUSR CTLRGN(TPC3) TRANID(TPC3))
            IF          COND((&SSNWHSED *GT (&STRWRHSD + 500)) & +
                          (&SSNWHSED *LE (&STRWRHSD + 600))) +
                          THEN(STRCICSUSR CTLRGN(TPC4) TRANID(TPC4))
            IF          COND((&SSNWHSED *GT (&STRWRHSD + 600)) & +
                          (&SSNWHSED *LE (&STRWRHSD + 700))) +
                          THEN(STRCICSUSR CTLRGN(TPC5) TRANID(TPC5))
            IF          COND((&SSNWHSED *GT (&STRWRHSD + 700)) & +
                          (&SSNWHSED *LE (&STRWRHSD + 800))) +
                          THEN(STRCICSUSR CTLRGN(TPC6) TRANID(TPC6))
            IF          COND((&SSNWHSED *GT (&STRWRHSD + 800)) & +
                          (&SSNWHSED *LE (&STRWRHSD + 900))) +
                          THEN(STRCICSUSR CTLRGN(TPC7) TRANID(TPC7))
            IF          COND((&SSNWHSED *GT (&STRWRHSD + 900)) & +
                          (&SSNWHSED *LE (&STRWRHSD + 1000))) +
                          THEN(STRCICSUSR CTLRGN(TPC8) TRANID(TPC8))
ENDPGM
```

SECONDGM2: Sign-On Program

```
            PGM PARM(&E &C)
            DCL &E     *CHAR 10
            DCL &C    *CHAR 10
            DCL &WRHS *CHAR 4 VALUE('0001')
            DCL &DST *CHAR 2 VALUE('01')
            DCL         VAR(&QNAME) TYPE(*CHAR) LEN(10) VALUE(TEMPDTAQ)
            DCL         VAR(&QLIB) TYPE(*CHAR) LEN(10) +
                          VALUE(QTEMP)
            DCL         VAR(&FLDLEN) TYPE(*DEC) LEN(5 0) VALUE(6)
            DCL         VAR(&WAIT) TYPE(*DEC) LEN(5 0) VALUE(0)
            DCL         VAR(&VALUE) TYPE(*CHAR) LEN(6)
            CALL        PGM(QRCVDTAQ) PARM(&QNAME &QLIB &FLDLEN +
                          &VALUE &WAIT)
            SNDPGMMSG   MSG(&VALUE)
            CHGVAR      VAR(&WRHS)     VALUE(%SST(&VALUE 1 4))
            CHGVAR      VAR(&DST)   VALUE(%SST(&VALUE 5 2))
            ?           SECONDGM1 WRHSNUM(&WRHS) DSTRCTNUM(&DST)


ENDIT:      ENDPGM
```

SECONDGM1: Sign-On Program

```
/*                                                              */
/*  This program:  Uses warehouse id & district id passed from cmd  */
/*                  file prompt.                                */
/*                 Calls the COBOL programs to execute the      */
/*                   transactions.                              */

            PGM  PARM(&SSNWHSE &SSNDSTRCT)
```

```
            DCL        VAR(&SSNWHSE) TYPE(*CHAR) LEN(4)
            DCL        VAR(&SSNDSTRCT) TYPE(*CHAR) LEN(2)
            DCL        VAR(&E) TYPE(*CHAR) LEN(10)
            DCL        VAR(&C) TYPE(*CHAR) LEN(10)

            DCL        VAR(&JOB) TYPE(*CHAR) LEN(10)
            DCL        VAR(&QNAME) TYPE(*CHAR) LEN(10) VALUE(INITDTAQ)
            DCL        VAR(&QLIB) TYPE(*CHAR) LEN(10) +
                         VALUE(TPCCCICS)
            DCL        VAR(&VALUE) TYPE(*CHAR) LEN(6)
            DCL        VAR(&VALUE1) TYPE(*DEC) LEN(6 0)
            DCL        VAR(&FLDLEN) TYPE(*DEC) LEN(5 0) VALUE(6)
            DCL        VAR(&WAIT) TYPE(*DEC) LEN(5 0) VALUE(0)
            DCL        VAR(&DEV) TYPE(*CHAR) LEN(10)
            DCL        VAR(&MODE) TYPE(*CHAR) LEN(8)
            DCL        VAR(&OBJLIB) TYPE(*CHAR) LEN(10) +
                         VALUE(TPCCKVCLNT)

            MONMSG     MSGID(CPF2103) /* Library already exists in +
                         library list */

            MONMSG     MSGID(CPF2451) /* The message queue is +
                         allocated to someone else */

            ADDLIBLE   LIB(&OBJLIB) /* TPC-C data base library */
            CHGCURLIB  CURLIB(&OBJLIB)

            RTVJOBA    JOB(&JOB)
            MONMSG     MSGID(CPF0000)

SNDPGMMSG &JOB
SNDPGMMSG &SSNWHSE
SNDPGMMSG &SSNDSTRCT
            CALL     PGM(&OBJLIB/ATPCCCICS) PARM(&E &C &JOB &SSNWHSE +
                       &SSNDSTRCT)
            MONMSG     MSGID(CEE9901) EXEC(DLYJOB DLY(9999))



            ENDPGM
```

APTCCCICS: Main TPCC Program

```
      PROCESS NOTRUNC NORANGE.
      IDENTIFICATION DIVISION.

        PROGRAM-ID. ATPCCCICS.
        AUTHOR. P/P COC
        INSTALLATION. IBM-ROCHESTER.
        DATE-WRITTEN.
        DATE-COMPILED.
      ****************************************************************
      *   C  H  A  N  G  E  S     &     U  P  D  A  T  E  S
      *
      *   LATEST CHANGES LISTED FIRST
      *
      *   MM/DD/YY   AUTHOR NAME          LINES CHANGED/ADDED: NNN
      *
      *
      *     DESCRIPTION OF CHANGE:
      *
      *
      ****************************************************************
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
         SOURCE-COMPUTER. IBM-AS400.
         OBJECT-COMPUTER. IBM-AS400.
         INPUT-OUTPUT SECTION.
         FILE-CONTROL.
             SELECT DISPLAY-FILE
                 ASSIGN TO WORKSTATION-ATPCCDSPF
                 ORGANIZATION IS TRANSACTION
                 CONTROL-AREA IS TRANSACTION-CONTROL-AREA
                 ACCESS MODE IS SEQUENTIAL.
         DATA DIVISION.
         FILE SECTION.
         FD DISPLAY-FILE
            LABEL RECORDS ARE OMITTED.
         01 DISPLAY-RECORD.

            COPY DDS-ALL-FORMATS OF ATPCCDSPF.

          05 DISPLAY-OUT REDEFINES ATPCCDSPF-RECORD.
               06    OUTPUT-FMT-NUM        PIC XX.
               06    TOTAMT                PIC S9(10)V9(2).
               06    OENTTM                PIC S9(6).
               06    CLAST                 PIC X(16).
               06    CCREDT                PIC X(2).
               06    CDCT                  PIC S99V99.
               06    OID                   PIC S9(8).
               06    OLINES                PIC S9(2).
               06    WTAX                  PIC S99V99.
               06    DTAX                  PIC S99V99.
               06  OUTPUT-TABLE.
                   15 OUTPUT-LINE OCCURS  15 TIMES.
                      20 OUTPUT-INAME       PIC X(24).
                      20 OUTPUT-STQTY       PIC s9(3).
                      20 OUTPUT-BORG        PIC  X(1).
                      20 OUTPUT-IPRICE      PIC s9(3)V9(2).
                      20 OUTPUT-OLAMNT      PIC s9(5)V9(2).
            05 NEWORDOUT2 REDEFINES DISPLAY-OUT.
               06  OUTPUT-FMT-NUM-2         PIC XX.
               06  CID2                     PIC X(4).
               06  CLAST                    PIC X(16).
               06  CCREDT                   PIC X(2).
               06  OID                      PIC S9(8).
          05 NEWORDOUT-TEMP REDEFINES DISPLAY-OUT.
               06    FILLER                PIC XX.
               06    TOTAL                 PIC S9(10)V9(2).
               06    FILLER                PIC S9(6).
               06    FILLER                PIC X(16).
               06    FILLER                PIC X(2).
               06    CDCT-TEMP             PIC SV9999.
               06    FILLER                PIC S9(8).
               06    FILLER                PIC S9(2).
               06    WTAX-TEMP             PIC SV9999.
               06    DTAX-TEMP             PIC SV9999.

            05 PAYMENT-OUTDSP REDEFINES ATPCCDSPF-RECORD.
               06 OUTPUT-FMT-NUM-3       PIC XX.
               06 hdate     PIC s9(8).
               06 htime     pic s9(6).
               06 WADDR1    PIC X(20).
               06 DADDR1    PIC X(20).
               06 WADDR2    PIC X(20).
               06 DADDR2    PIC X(20).
               06 WH-DATA.
                  08 WCITY     PIC X(20).
                  08 WSTATE    PIC X(02).
                  08 WZIP      PIC X(10).
               06 DST-DATA.
                  08 DCITY     PIC X(20).
                  08 DSTATE    PIC X(02).
                  08 DZIP      PIC X(10).
               06 CID          PIC X(4).
               06 CIDN REDEFINES CID  PIC 9(4).
               06 PAYMENT-CUST-INFO.
                  08 CFIRST    PIC X(16).
                  08 CINIT     PIC X(02).
                  08 CLAST     PIC X(16).
                  08 CLDATE    PIC S9(8).
                  08 CADDR1    PIC X(20).
                  08 CCREDT    PIC X(02).
                  08 CADDR2    PIC X(20).
               06 CDCT         PIC S99V99.
               06 PAYMENT-CUST-INFO2.
                  08 CCITY     PIC X(20).
                  08 CSTATE    PIC X(02).
                  08 CZIP      PIC X(10).
                  08 CPHONE    PIC X(19).
               06 CBAL      PIC S9(10)V9(2).
               06 PAYMNT    PIC S9(4)V9(2).
               06 CCRDLM    PIC S9(10)V9(2).
               06 MISC-CDATA.
                  08  CDAT1    PIC X(50).
                  08  CDAT2    PIC X(50).
                  08  CDAT3    PIC X(50).
                  08  CDAT4    PIC X(50).
            05 PAYMENT-OUTDSP2 REDEFINES PAYMENT-OUTDSP.
               06 OUTPUT-FMT-NUM-4       PIC XX.
               06 HDATE     PIC S9(8).
               06 HTIME     PIC S9(6).
               06 WADDR1    PIC X(20).
               06 DADDR1    PIC X(20).
               06 WADDR2    PIC X(20).
               06 DADDR2    PIC X(20).
               06 WH-DATA.
                  08 WCITY     PIC X(20).
                  08 WSTATE    PIC X(02).
                  08 WZIP      PIC X(10).
               06 DST-DATA.
                  08 DCITY     PIC X(20).
                  08 DSTATE    PIC X(02).
                  08 DZIP      PIC X(10).
               06 CID          PIC X(4).
               06 PAYMENT-CUST-INFO.
                  08 CFIRST    PIC X(16).
                  08 CINIT     PIC X(02).
                  08 CLAST     PIC X(16).
                  08 CLDATE    PIC S9(8).
                  08 CADDR1    PIC X(20).
                  08 CCREDT    PIC X(02).
                  08 CADDR2    PIC X(20).
               06 CDCT         PIC S99V99.
               06 PAYMENT-CUST-INFO2.
                  08 CCITY     PIC X(20).
                  08 CSTATE    PIC X(02).
                  08 CZIP      PIC X(10).
                  08 CPHONE    PIC X(19).
               06 CBAL      PIC S99999V99.
               06 CCRDLM    PIC S99999V99.

            05  ORDSTS-OUTDSP2 REDEFINES ATPCCDSPF-RECORD.
               06 ORDSTS-FMT-NUM         PIC XX.
               06 OCID                   PIC X(4).
               06 OCIDN REDEFINES OCID   PIC 9(4).
               06 CFIRST                 PIC X(16).
               06 CINIT                  PIC X(2).
               06 CLAST                  PIC X(16).
               06 CBAL                   PIC S9(10)V9(2).
               06 OID                    PIC S9(8).
               06 OENTDT                 PIC S9(8).
               06 OENTTM                 PIC S9(6).
               06 OCARID                 PIC X(2).
               06 DO-LINE OCCURS 15 TIMES.
                  07 DO-OLSPWH           PIC X(4).
                  07 DO-OLIID            PIC X(6).
                  07 DO-OLQTY            PIC S9(3).
                  07 DO-OLAMNT           PIC S9(5)V9(2).
                  07 DO-OLDLVD           PIC S9(8).

          01 SNDTO-SRVR.
             03 TXNTYPE      PIC X.
             03 SNDJOBNM     PIC X(10).
             03 TXNDATA      PIC X(219).


        WORKING-STORAGE SECTION.

      * CICS DEFINITIONS
          01 COMMAREAD PIC X(801).

            03 TRANS-TYPE.
              06 REGION         PIC S9(4) COMP-4.
              06 TTYPE          PIC S9(4) COMP-4.
              06 PRIORITY       PIC S9(4) COMP-4.
              06 LOAD-BALANCE   PIC S9(9) COMP-4.

            03 DATA-PTR   USAGE POINTER.

            03 DATA-TYPE.
              06 DATA-LEN       PIC S9(4) COMP-4.
              06 RESERVED       PIC X(10).
              06 DATA-DAT       PIC X(750).

            03 ERR-PTR USAGE POINTER.

            03 ERR-TYPE.
              06 BYTES-PROV     PIC S9(4) COMP-4.
              06 BYTES-AVAIL    PIC S9(4) COMP-4.
              06 EXCEPT-ID      PIC X(7).
              06 RESERVED       PIC X(1).


            01 DUMMYB     PIC X(40).

            01 PAYMNTT                PIC S9(4)V9(2).

            01 SPACES-STRUCT          PIC X(230) VALUE SPACES.
            01 SPACE-LINE.
                  5 FILLER             PIC X(4) VALUE SPACES.
                  5 FILLER             PIC X(6) VALUE SPACES.
                  5 FILLER             PIC S9(3) VALUE ZEROS.
            01 FLAG PIC X.

            01  TWO-SPACES.
                03 FILLER        PIC XX VALUE SPACES.
```

```
    01  TWO-BIN-SPACES REDEFINES TWO-SPACES.
        03 BIN-SPACES    PIC 9(4) BINARY.
    01  TEMP-AMT  PIC 9V9999 COMP-3.
    01  DISPLAY-IN              PIC X(222) VALUE SPACES.
    01  DISPLAY-NEWORDIN REDEFINES DISPLAY-IN.
        06 CDID                 PIC S9(2).
        06  DIST-NUM REDEFINES CDID   PIC 99.
        06 CID                  PIC X(4).
        06 INPUT-TABLE.
          08 INPUT-LINE OCCURS 16 TIMES.
             10 INPUT-OLSPWH       PIC X(4).
             10 INPUT-OLIID        PIC X(6).
             10 INPUT-OLQTY        PIC 9(3).
        06 NEWORDER-DISTRICT       PIC X(2).
        06 NEWORDER-WID            PIC X(4).


    01  DISPLAY-ORDSTSIN REDEFINES DISPLAY-IN.
        06  HIDEIT                 PIC XX.
        06  ODID                   PIC S9(2).
        06  OCID                   PIC X(4).
        06  CLAST                  PIC X(16).
        06  ORDSTS-DISTRICT        PIC X(2).
        06  ORDSTS-WAREHOUSE       PIC X(4).

    01  DISPLAY-PAYMNTIN2.
            06  HIST-DATA.
                07  DID                    PIC S9(2).
                07  CID                    PIC X(4).
                07  CWID                   PIC X(4).
                07  CDID                   PIC S9(2).
            06  HDATA                  PIC X(24).
            06  MISC-HDATA REDEFINES HDATA.
                08  CLAST                  PIC X(16).
                08  PAYMNT                 PIC S9(4)V9(2).
                08  FILLER                 PIC X(2).
            06  HWID2                  PIC X(4).
            06  HDATE                  PIC S9(8).
            06  HTIME                  PIC S9(6).
            06  TEMP-PAYMENT           PIC S9(7)V9(2).
            06  PAYMENT-DISTRICT       PIC X(2).
            06  PAYMENT-WAREHOUSE      PIC X(4).

   01 SLDATA.
       05 SLDATA-TERMINAL.
          06 FILLER               PIC XX.
          06 SLDATA-WID           PIC XXXX.
          06 FILLER               PIC XX.
          06 FILLER               PIC XX.
       05 SLDATA-THRSH            PIC S9(2).
       05 SLDATA-DIST             PIC X(2).

   01 SLRXDATA.
       05 SLRX-TERMINAL           PIC X(10).
       05 SLRX-BLWSTK             PIC S9(3).
       05 FILLER                  PIC X(1).

*****************************************************************


   01  FORMAT-ID.
       05  FILLER          PIC X(6) VALUE "DSPREC".
       05  FMT-NUM         PIC X(2) VALUE SPACES.
       05  NUMBER-OF-ITEMS REDEFINES FMT-NUM
                           PIC 99.
    01 I                                  PIC S999 BINARY.
    01 J                                  PIC S999 BINARY.
    01 NUMBER-OF-ORDERLINES               PIC S999 BINARY.

    01 TORD    PIC X(1) VALUE "t".
    01 NEWORDTM PIC X(6) VALUE SPACES.
    01 CLDATEDT PIC X(8) VALUE SPACES.
    01 OLDLVDT  PIC X(8) VALUE SPACES.
    01 PAYDTTIM.
       05 PAYDT PIC X(8) VALUE SPACES.
       05 PAYTM PIC X(6) VALUE SPACES.
    01 ORDDTTIM.
       05 ORDDT PIC X(8) VALUE SPACES.
       05 ORDTM PIC X(6) VALUE SPACES.

    01 DATEINT.
       06 YY          PIC 9(2).
       06 MMDD        PIC 9(4).

   01 DATETIME.
      05 TPCDATE.
         06  TPCMD             PIC 9(4).
         06  TPCCEN            PIC 9(2) VALUE 19.
         06  TPCYEAR           PIC 9(2).
      05 CURTIME.
         06 CURTIME6            PIC 9(6).
         06 FILLER              PIC 9(2).

  01  TRANSACTION-CONTROL-AREA.
       05  FUNCTION-KEY          PIC XX.
       05  TERMINAL-ID           PIC X(10).
       05  FORMAT-NAME           PIC X(10).

   01 PRM-WH                   PIC XXXX.
   01 PRM-DIST                 PIC XX.

   01 PRM-WH-INT                        PIC 9999.
   01 TTEMP-PRM-WH                      PIC 9999.
   01 TEMP-PRM-WH REDEFINES TTEMP-PRM-WH PIC XXXX.


   01 SNDJOBNM     PIC X(10).


*****************************************
* THE DATA STRUCTURES USED BY THE SERVER *
*****************************************


  01   TRANSACTION-INPUT.
         06    TRANSACTION-TYPE    PIC X.
         06    CLIENT-INPUT        PIC X(195).
         06 NEWORD-I REDEFINES CLIENT-INPUT.
           08 CWID                 PIC S9(4) COMP-4.
           08 CDID                 PIC S9(2) COMP-4.
           08 CID                  PIC S9(6) COMP-4.
           08 NUMBER-OF-ITEMS      PIC 9(3) COMP-3.
           08 INPUT-LINE1 OCCURS 16 TIMES.
             10 OLSPWHI             PIC S9(4) COMP-4.
             10 OLIIDI              PIC S9(6) COMP-4.
             10 OLQTYI              PIC S9(3) COMP-3.
         06 PAYMENT-I REDEFINES CLIENT-INPUT.
           08 PAYMENT-TYPE         PIC X.
*                                 Payment type = C if by CID
*                                 Payment type = L if by CLAST
           08 HISTORY-DATA.
            09 PAY-DIST-KEY.
             10 DID                PIC S9(2) COMP-4.
             10 WID                PIC S9(4) COMP-4.
            09 PAY-CUST-KEY.
             10 CID                PIC S9(6) COMP-4.
             10 CDID               PIC S9(2) COMP-4.
             10 CWID               PIC S9(4) COMP-4.
            09 AMOUNT              PIC S9(5)V99 COMP-3.
           08 CLAST                PIC X(16).
           08 FILLER               PIC X(157).
         06 ORDSTS-I REDEFINES CLIENT-INPUT.
           08 ORDSTS-TYPE          PIC X.
           08 ORD-CUST-KEY.
            09 CID                 PIC S9(6) COMP-4.
            09 DID                 PIC S9(2) COMP-4.
            09 WID                 PIC S9(4) COMP-4.
           08 CLAST                PIC X(16).
           08 FILLER               PIC X(165).
         06 STKLVL-I REDEFINES CLIENT-INPUT.
           08 WID                  PIC S9(4) COMP-4.
           08 DID                  PIC S9(2) COMP-4.
           08 THRESHOLD            PIC S9(2) COMP-4.
           08 BLWSTK               PIC S9(4) COMP-4.
         06 DLVRY-I REDEFINES CLIENT-INPUT.
           08 WID                  PIC S9(4) COMP-4.
           08 CARRIER              PIC XX.
           08 D-TIME               PIC S9(8).
           08 D-DATE               PIC S9(6).

   01 PRM-DTAQLIB              PIC X(10).

*****************************************************************
*  ARRAY DECLARATION FOR HANDLING ORDER LINE RECORDS
*****************************************************************


   01 OLINEINF.
          07 DO-OLSPWH1           PIC S9(4) COMP-4.
          07 DO-OLIID1            PIC S9(6) COMP-4.
          07 DO-OLQTY1            PIC S9(3) COMP-3.
          07 DO-OLAMNT1           PIC S9(5)V9(2) COMP-3.
          07 DO-OLDLVD1           PIC X(8).

   01 DOLINE.
          07 DO-OLSPWH2          PIC X(4).
          07 DO-OLIID2           PIC X(6).
          07 DO-OLQTY2           PIC S9(3).
          07 DO-OLAMNT2          PIC S9(5)V9(2).
          07 DO-OLDLVD2          PIC S9(8).


   01 DOLINE-NULL.
          07 DO-OLSPWH3          PIC X(4) VALUE SPACES.
          07 DO-OLIID3           PIC X(6) VALUE SPACES.
          07 DO-OLQTY3           PIC S9(3) VALUE 0.
          07 DO-OLAMNT3          PIC S9(5)V9(2) VALUE 0.
          07 DO-OLDLVD3          PIC S9(8) VALUE 0.

    01 DQDATA.
       05 DQDATA-WHS  PIC X(4).
       05 DQDATA-CARR PIC X(2).
       05 DQDATA-BTIM PIC X(8).
       05 DQDATA-fill PIC X(6).


   01 RETURN-DATA.

   05 NEWORD-O.
         06    NEWORD-RESULT         PIC X.
         06    OENTTM                PIC S9(6).
         06    CLAST                 PIC X(16).
         06    CCREDT                PIC X(2).
         06    CDCT                  PIC S999V99 COMP-3.
         06    OID                   PIC S9(8) COMP-4.
         06    WTAX                  PIC S9V9999 COMP-3.
         06    DTAX                  PIC S9V9999 COMP-3.
         06    TOTAMT                PIC S9(9)V9(2) COMP-3.
         06  OUTPUT-TABLE.
           10 OUTPUT-LINE1 OCCURS  15 TIMES.
              15 OUTPUT-STQTY1        PIC S9(4) COMP-4.
              15 OUTPUT-BORG1         PIC  X(1).
              15 OUTPUT-ITEM-INFO.
                20 OUTPUT-INAME1       PIC X(24).
                20 OUTPUT-IPRICE1      PIC S9(3)V9(2) COMP-3.
              15 OUTPUT-OLAMNT1      PIC S9(5)V9(2) COMP-3.
         06   OENTTOD                PIC X(8).
         06 FILLER PIC X(108).
   01  PAYFROM-SRVR  REDEFINES RETURN-DATA.
       05 PAYDATABACK PIC X(527).
       05 PAYMNT-O REDEFINES PAYDATABACK.
         06 OUTPUT-FMT-NUM1-3      PIC XX.
*                                    17 = bad credit
*                                    18 = no cid or clast
*                                    19 = good credit
*                                    88 = bad last name
         06 HDT.
           08 HOTOD    PIC X(8).
           08 HTIME    pic S9(6).
         06 WH-DATA.
           08 WADDR1   PIC X(20).
           08 WADDR2   PIC X(20).
           08 WCITY    PIC X(20).
           08 WSTATE   PIC X(02).
           08 WZIP     PIC X(9).
         06 DST-DATA.
           08 DADDR1   PIC X(20).
           08 DADDR2   PIC X(20).
           08 DCITY    PIC X(20).
           08 DSTATE   PIC X(02).
           08 DZIP     PIC X(9).
         06 CID        PIC S9(6) COMP-4.
         06 PAYMENT-CUST-INFO.
           08 CFIRST   PIC X(16).
           08 CINIT    PIC X(02).
           08 CLAST    PIC X(16).
           08 CLDATE   PIC S9(8).
           08 CADDR1   PIC X(20).
           08 CCREDT   PIC X(02).
           08 CADDR2   PIC X(20).
           08 CDCT     PIC S9V9999 COMP-3.
           08 CCITY    PIC X(20).
           08 CSTATE   PIC X(02).
           08 CZIP     PIC X(9).
           08 CPHONE   PIC X(16).
           08 CBAL     PIC S9(11)V9(2) COMP-3.
           08 CCRDLM   PIC S9(11)V9(2) COMP-3.
         06 MISC-CDATA.
           08  CDAT1    PIC X(50).
           08  CDAT2    PIC X(50).
           08  CDAT3    PIC X(50).
           08  CDAT4    PIC X(50).


   01  OS-FROM-SRVR  REDEFINES RETURN-DATA.
       05 OSDATABACK PIC X(512).
       05  ORDSTS-O REDEFINES OSDATABACK.
```

```
        06 ORDSTS-FMT-NUM1       PIC XX.
        06 OCID                  PIC S9(6) COMP-4.
        06 CUSTINFO.
           08 CFIRST                 PIC X(16).
           08 CINIT                  PIC X(2).
           08 CLAST                  PIC X(16).
        06 CBAL                  PIC S9(11)V9(2) COMP-3.
        06 ORDERINFO.
           08 OID                    PIC S9(8) COMP-4.
           08 ODT.
              10 OENTTOD                PIC X(8).
              10 OENTTM                 PIC 9(6).
           08 OCARID                 PIC X(2).
           08 OLINENBR               PIC S9(2) COMP-4.
        06 OLINEINFO OCCURS 15 TIMES.
          07 DO-OLSPWH           PIC S9(4) COMP-4.
          07 DO-OLIID            PIC S9(6) COMP-4.
          07 DO-OLQTY            PIC S9(3) COMP-3.
          07 DO-OLAMNT           PIC S9(5)V9(2) COMP-3.
          07 DO-OLDLVTOD         PIC X(8).
     01  SL-FROM-SRVR  REDEFINES RETURN-DATA.
         05 SLDATABACK PIC X(14).
         05 STKLVL-O REDEFINES SLDATABACK.
            06 FILLER                 PIC X(2).
            06 BLWSTK                 PIC S9(2) COMP-4.

     01 DL-FROM-SRVR REDEFINES RETURN-DATA.
          05 DLVRY-O.
            06 FILLER                 PIC X(2).
            06 DELIVERY-STATUS        PIC X.


      01 DUMMYA   PIC X(40).
      01 FIRSTTIME       PIC X(1) VALUE "Y".

*****************************************
* THE DATA STRUCTURES USED BY THE SERVER *
*****************************************

*****************************************************************
*  LINKAGE SECTION FOR DATA PASSED TO THE PROGRAM
*****************************************************************
 LINKAGE SECTION.

    01 PRM-JOBNAM              PIC X(10).
    01 PRM-WH1                 PIC XXXX.
    01 PRM-DIST1               PIC XX.


 PROCEDURE DIVISION USING PRM-JOBNAM, PRM-WH1, PRM-DIST1.

   MAIN-LINE-ROUTINE.

       PERFORM SET-UP-ROUTINE.
       PERFORM CMD-KEY-ROUTINE THRU
               CMD-KEY-ROUTINE-EXIT
         UNTIL FUNCTION-KEY = "03".

       CLOSE DISPLAY-FILE.
       STOP RUN.


*****************************************************************
*  CMD-KEY ROUTINE PERFORMS ALL DISPLAY FILE I/O AND
*  DETERMINES WHAT TRANSACTION TYPE TO PROCESS.
*****************************************************************

 CMD-KEY-ROUTINE.
        IF FUNCTION-KEY = "03"
            GO TO CMD-KEY-ROUTINE-EXIT
        END-IF.

* WRITE SCREEN I/O RECORD WITH THE FOLLOWING WRITE STATEMENT

     WRITE DISPLAY-RECORD FORMAT IS FORMAT-ID.

*  READ SCREEN I/O, PAYMENT RECORD IS DONE DIFFERETLY

     IF FORMAT-ID = "DSPREC88"
          READ DISPLAY-FILE INTO DISPLAY-PAYMNTIN2
               FORMAT IS FORMAT-ID
       ELSE
          READ DISPLAY-FILE INTO DISPLAY-IN
               FORMAT IS FORMAT-ID.

*****************************************************************
*Check for transfer job request
* --TESTS FEEDBACK AREA FOR CA-KEY--
*****************************************************************



        IF FUNCTION-KEY = "00"
           IF  FORMAT-NAME = "DSPREC00"
             PERFORM HANDLE-NEWORD THRU HANDLE-NEWORD-EXIT

           ELSE
           IF FORMAT-NAME = "DSPREC88"
              PERFORM HANDLE-PAYMNT THRU HANDLE-PAYMNT-EXIT
              MOVE OUTPUT-FMT-NUM-3 TO FMT-NUM

           ELSE
           IF FORMAT-NAME = "DSPREC22"
             PERFORM HANDLE-DLVRY THRU HANDLE-DLVRY-EXIT
              MOVE "23" TO FMT-NUM

           ELSE
           IF FORMAT-NAME = "DSPREC30"
            PERFORM HANDLE-ORDSTS THROUGH HANDLE-ORDSTS-EXIT
               MOVE ORDSTS-FMT-NUM  TO FMT-NUM


           ELSE
           IF FORMAT-NAME = "DSPREC50"
             PERFORM HANDLE-STKLVL THRU HANDLE-STKLVL-EXIT
                 MOVE "51" TO FMT-NUM

           ELSE
           IF  FORMAT-NAME = "DSPREC99"

                MOVE "00" TO FMT-NUM
                PERFORM HANDLE-NEWORD THRU HANDLE-NEWORD-EXIT

           END-IF.

        IF FUNCTION-KEY = "01"
           MOVE "00" TO FMT-NUM
           MOVE PRM-WH TO CWID OF DSPREC00-O
           MOVE TPCDATE TO OENTDT OF DSPREC00-O
        ELSE

        IF FUNCTION-KEY = "02"
           MOVE "88" TO FMT-NUM
           MOVE PRM-WH TO WID OF DSPREC88-O
           ELSE
           IF FUNCTION-KEY = "04"
              MOVE "22" TO FMT-NUM
              MOVE PRM-WH TO WID OF DSPREC22-O
           ELSE
           IF FUNCTION-KEY = "05"
              MOVE "30" TO FMT-NUM
              MOVE PRM-WH TO OWID OF DSPREC30-O
           ELSE
           IF FUNCTION-KEY = "06"
              MOVE "50"      TO FMT-NUM
              MOVE PRM-WH    TO DWID OF DSPREC50-O
              MOVE PRM-DIST1 TO DID  OF DSPREC50-O.
CMD-KEY-ROUTINE-EXIT.
     EXIT.

HANDLE-NEWORD.
    MOVE PRM-WH TO NEWORDER-WID OF DISPLAY-NEWORDIN.
         MOVE PRM-WH TO CWID OF NEWORD-I.
         MOVE CDID OF DISPLAY-NEWORDIN TO CDID OF NEWORD-I.
         MOVE CID OF DISPLAY-NEWORDIN TO CID OF NEWORD-I.
         MOVE SPACE-LINE TO INPUT-LINE(16).
         PERFORM VARYING I FROM
              1 BY 1 UNTIL INPUT-LINE(I) = SPACE-LINE
          IF INPUT-OLSPWH(I) = SPACES MOVE "62" TO HIDEME OF
           DSPREC62
           MOVE "62" TO FMT-NUM OF FORMAT-ID
           GO TO HANDLE-NEWORD-EXIT
          END-IF
                MOVE INPUT-OLSPWH(I) TO
              OLSPWHI(I)

          IF INPUT-OLIID(I) = SPACES MOVE "62" TO HIDEME OF
           DSPREC62
           MOVE "62" TO FMT-NUM OF FORMAT-ID
           GO TO HANDLE-NEWORD-EXIT
          END-IF
                MOVE INPUT-OLIID(I) TO
              OLIIDI(I)
          IF INPUT-OLQTY(I) = 0 MOVE "62" TO HIDEME OF
           DSPREC62
           MOVE "62" TO FMT-NUM OF FORMAT-ID
           GO TO HANDLE-NEWORD-EXIT
          END-IF
          MOVE INPUT-OLQTY(I) TO
               OLQTYI(I)
         END-PERFORM.
         SUBTRACT 1 FROM I GIVING NUMBER-OF-ORDERLINES.
         MOVE NUMBER-OF-ORDERLINES TO
                   NUMBER-OF-ITEMS OF NEWORD-I.
                   SUBTRACT I FROM 16 GIVING J.
         PERFORM J TIMES
           ADD 1 TO I GIVING I
           IF INPUT-LINE(I) NOT EQUAL SPACE-LINE
    MOVE "ID" TO FMT-NUM OF FORMAT-ID, OUTPUT-FMT-NUM-2
              GO TO HANDLE-NEWORD-EXIT
           END-IF
           END-PERFORM.

    MOVE "N" TO TXNTYPE OF SNDTO-SRVR.
    MOVE PRM-JOBNAM TO SNDJOBNM OF SNDTO-SRVR.
    MOVE NEWORD-I TO TXNDATA OF SNDTO-SRVR.

    SET DATA-PTR TO ADDRESS OF DATA-TYPE.
    SET ERR-PTR  TO ADDRESS OF ERR-TYPE.

    MOVE  1 TO REGION.
    MOVE 77 TO TTYPE.
    MOVE  0 TO PRIORITY.
    MOVE  0 TO LOAD-BALANCE.

    MOVE 230              TO DATA-LEN OF DATA-TYPE.
    MOVE SNDTO-SRVR(1:230) TO DATA-DAT OF DATA-TYPE.

    EXEC CICS LINK PROGRAM('AEGTMRUN') COMMAREA(COMMAREAD)
          LENGTH(801) END-EXEC.

    MOVE DATA-DAT(1:670) TO NEWORD-O.

    MOVE NUMBER-OF-ORDERLINES TO OLINES OF DISPLAY-OUT
          IF NEWORD-RESULT = "G"
          MOVE CLAST  OF NEWORD-O TO CLAST  OF DISPLAY-OUT
          MOVE CCREDT OF NEWORD-O TO CCREDT OF DISPLAY-OUT
          MOVE OID    OF NEWORD-O TO OID    OF DISPLAY-OUT
          MOVE "t" TO TORD
          CALL "getdttmstr" using OENTTOD OF NEWORD-O,
                NEWORDTM , BY VALUE TORD

          MOVE NEWORDTM           TO OENTTM OF DISPLAY-OUT
          MOVE CDCT   OF NEWORD-O TO CDCT   OF DISPLAY-OUT
          MOVE WTAX   OF NEWORD-O TO WTAX   OF DISPLAY-OUT
          MOVE DTAX   OF NEWORD-O TO DTAX   OF DISPLAY-OUT
          MOVE TOTAMT OF NEWORD-O TO TOTAMT OF DISPLAY-OUT
          PERFORM VARYING I FROM
               1 BY 1 UNTIL I > NUMBER-OF-ORDERLINES
             MOVE OUTPUT-STQTY1(I) TO
                  OUTPUT-STQTY(I)
             MOVE OUTPUT-BORG1(I)  TO
                  OUTPUT-BORG(I)
             MOVE OUTPUT-INAME1(I) TO
                  OUTPUT-INAME(I)
             MOVE OUTPUT-IPRICE1(I) TO
                  OUTPUT-IPRICE(I)
             MOVE OUTPUT-OLAMNT1(I) TO
                  OUTPUT-OLAMNT(I)
           END-PERFORM
    MOVE I TO NUMBER-OF-ITEMS OF FORMAT-ID
    MOVE FMT-NUM TO OUTPUT-FMT-NUM
    SUBTRACT CDCT-TEMP FROM 1 GIVING TEMP-AMT
    MULTIPLY TOTAL BY TEMP-AMT GIVING TOTAL ROUNDED
    ADD WTAX-TEMP TO 1 GIVING TEMP-AMT
    ADD DTAX-TEMP TO TEMP-AMT GIVING TEMP-AMT
    MULTIPLY TOTAL BY TEMP-AMT GIVING TOTAL ROUNDED
    ELSE IF NEWORD-RESULT = "R"
          MOVE CLAST  OF NEWORD-O TO CLAST  OF NEWORDOUT2
          MOVE CCREDT OF NEWORD-O TO CCREDT OF NEWORDOUT2
    MOVE CID OF DISPLAY-NEWORDIN TO CID2
    MOVE OID OF NEWORD-O TO OID OF NEWORDOUT2
    MOVE "RB" TO FMT-NUM OF FORMAT-ID, OUTPUT-FMT-NUM-2
    IF NEWORD-RESULT = "I"
    MOVE "ID" TO FMT-NUM OF FORMAT-ID, OUTPUT-FMT-NUM-2
    END-IF
    ELSE
         CALL "DLYFOREVER"
    END-IF.
HANDLE-NEWORD-EXIT.
    ADD 0 TO I GIVING I.
      EXIT.

HANDLE-PAYMNT.
    MOVE PRM-WH1  TO PAYMENT-WAREHOUSE
                     OF DISPLAY-PAYMNTIN2
          MOVE PRM-WH1 TO WID OF PAYMENT-I
          MOVE DID OF DISPLAY-PAYMNTIN2 TO DID OF PAYMENT-I
```

```
           MOVE CDID OF DISPLAY-PAYMNTIN2 TO CDID OF PAYMENT-I
       IF CID OF DISPLAY-PAYMNTIN2 EQUAL "    " MOVE "0000" TO
          CID OF DISPLAY-PAYMNTIN2
       END-IF
       IF CID OF DISPLAY-PAYMNTIN2 NOT EQUAL "0000"
          MOVE "C" TO PAYMENT-TYPE
       ELSE
          IF CLAST OF DISPLAY-PAYMNTIN2 NOT = SPACES
             MOVE "L" TO PAYMENT-TYPE
          ELSE
             MOVE "18" TO OUTPUT-FMT-NUM-3
             MOVE "18" TO HIDEME OF DSPREC18
             GO TO HANDLE-PAYMNT-EXIT
          END-IF
       END-IF
           MOVE CID OF DISPLAY-PAYMNTIN2 TO CID OF PAYMENT-I
           MOVE CWID OF DISPLAY-PAYMNTIN2 TO CWID OF PAYMENT-I
           MOVE PAYMNT OF DISPLAY-PAYMNTIN2 TO
                AMOUNT OF PAYMENT-I
           MOVE CLAST OF DISPLAY-PAYMNTIN2 TO
                CLAST OF PAYMENT-I

       MOVE "P"        TO TXNTYPE  OF SNDTO-SRVR.
       MOVE PRM-JOBNAM TO SNDJOBNM OF SNDTO-SRVR.
       MOVE PAYMENT-I  TO TXNDATA  OF SNDTO-SRVR.

       SET DATA-PTR TO ADDRESS OF DATA-TYPE.
       SET ERR-PTR  TO ADDRESS OF ERR-TYPE.

       MOVE  1 TO REGION.
       MOVE 88 TO TTYPE.
       MOVE  0 TO PRIORITY.
       MOVE  0 TO LOAD-BALANCE.

       MOVE 230              TO DATA-LEN OF DATA-TYPE.
       MOVE SNDTO-SRVR(1:230) TO DATA-DAT OF DATA-TYPE.

       EXEC CICS LINK PROGRAM('AEGTMRUN') COMMAREA(COMMAREAD)
               LENGTH(801) END-EXEC.

       MOVE DATA-DAT(1:670) TO PAYMNT-O.

           MOVE OUTPUT-FMT-NUM1-3 OF PAYMNT-O TO
                OUTPUT-FMT-NUM-3 OF PAYMENT-OUTDSP
           IF OUTPUT-FMT-NUM-3 OF PAYMENT-OUTDSP = "77"
           MOVE "77" TO HIDEME OF DSPREC77
           GO TO HANDLE-PAYMNT-EXIT.


           MOVE "b" TO TORD
           CALL "getdttmstr" using HOTOD   OF PAYMNT-O,
                   PAYDTTIM , BY VALUE TORD


           MOVE PAYDT              TO HDATE OF PAYMENT-OUTDSP
           MOVE PAYTM              TO HTIME OF PAYMENT-OUTDSP
           MOVE WADDR1 OF PAYMNT-O TO WADDR1 OF PAYMENT-OUTDSP
           MOVE WADDR2 OF PAYMNT-O TO WADDR2 OF PAYMENT-OUTDSP
           MOVE DADDR1 OF PAYMNT-O TO DADDR1 OF PAYMENT-OUTDSP
           MOVE DADDR2 OF PAYMNT-O TO DADDR2 OF PAYMENT-OUTDSP
           MOVE WCITY  OF PAYMNT-O TO WCITY  OF PAYMENT-OUTDSP
           MOVE WSTATE OF PAYMNT-O TO WSTATE OF PAYMENT-OUTDSP
           MOVE WZIP   OF PAYMNT-O TO WZIP   OF PAYMENT-OUTDSP
           MOVE DCITY  OF PAYMNT-O TO DCITY  OF PAYMENT-OUTDSP
           MOVE DSTATE OF PAYMNT-O TO DSTATE OF PAYMENT-OUTDSP
           MOVE DZIP   OF PAYMNT-O TO DZIP   OF PAYMENT-OUTDSP
           MOVE CID    OF PAYMNT-O TO CIDN   OF PAYMENT-OUTDSP
           MOVE CFIRST OF PAYMNT-O TO CFIRST OF PAYMENT-OUTDSP
           MOVE CINIT  OF PAYMNT-O TO CINIT  OF PAYMENT-OUTDSP
           MOVE CLAST  OF PAYMNT-O TO CLAST  OF PAYMENT-OUTDSP

           MOVE "d" TO TORD
           CALL "getdttmstr" using CLDATE  OF PAYMNT-O,
                   CLDATEDT , BY VALUE TORD

           MOVE CLDATEDT           TO CLDATE OF PAYMENT-OUTDSP
           MOVE CADDR1 OF PAYMNT-O TO CADDR1 OF PAYMENT-OUTDSP
           MOVE CADDR2 OF PAYMNT-O TO CADDR2 OF PAYMENT-OUTDSP
           MOVE CDCT   OF PAYMNT-O TO CDCT   OF PAYMENT-OUTDSP
           MOVE CCREDT OF PAYMNT-O TO CCREDT OF PAYMENT-OUTDSP
           MOVE CCITY  OF PAYMNT-O TO CCITY  OF PAYMENT-OUTDSP
           MOVE CSTATE OF PAYMNT-O TO CSTATE OF PAYMENT-OUTDSP
           MOVE CZIP   OF PAYMNT-O TO CZIP   OF PAYMENT-OUTDSP
           MOVE CPHONE OF PAYMNT-O TO CPHONE OF PAYMENT-OUTDSP
           MOVE CBAL   OF PAYMNT-O TO CBAL   OF PAYMENT-OUTDSP
           MOVE AMOUNT OF PAYMENT-I TO PAYMNT OF PAYMENT-OUTDSP
           MOVE CCRDLM OF PAYMNT-O TO CCRDLM OF PAYMENT-OUTDSP
           MOVE MISC-CDATA OF PAYMNT-O TO
                MISC-CDATA OF PAYMENT-OUTDSP.
HANDLE-PAYMNT-EXIT.
     EXIT.

HANDLE-DLVRY.
       MOVE OCARID OF DSPREC22-I TO CARRIER OF DLVRY-I.
       MOVE PRM-WH TO WID OF DLVRY-I.
       ACCEPT D-TIME OF DLVRY-I FROM TIME.

       MOVE "D"        TO TXNTYPE  OF SNDTO-SRVR.
       MOVE PRM-JOBNAM TO SNDJOBNM OF SNDTO-SRVR.
       MOVE DLVRY-I    TO TXNDATA  OF SNDTO-SRVR.

       SET DATA-PTR TO ADDRESS OF DATA-TYPE.
       SET ERR-PTR  TO ADDRESS OF ERR-TYPE.

       MOVE  1 TO REGION.
       MOVE 99 TO TTYPE.
       MOVE  0 TO PRIORITY.
       MOVE  0 TO LOAD-BALANCE.

       MOVE 230               TO DATA-LEN OF DATA-TYPE.
       MOVE SNDTO-SRVR(1:230) TO DATA-DAT OF DATA-TYPE.

       EXEC CICS LINK PROGRAM('AEGTMRUN') COMMAREA(COMMAREAD)
               LENGTH(801) END-EXEC.

HANDLE-DLVRY-EXIT.
     EXIT.

HANDLE-STKLVL.
       ACCEPT CURTIME FROM TIME.
       MOVE THRSH OF DSPREC50-I TO SLDATA-THRSH OF SLDATA.
       MOVE PRM-DIST1 TO SLDATA-DIST OF SLDATA.
       MOVE PRM-WH1   TO SLDATA-WID  OF SLDATA.
           MOVE SLDATA-WID   TO WID       OF STKLVL-I
           MOVE SLDATA-DIST  TO DID       OF STKLVL-I
           MOVE SLDATA-THRSH TO THRESHOLD OF STKLVL-I

       MOVE "S"        TO TXNTYPE  OF SNDTO-SRVR.
       MOVE PRM-JOBNAM TO SNDJOBNM OF SNDTO-SRVR.
       MOVE STKLVL-I   TO TXNDATA  OF SNDTO-SRVR.

       SET DATA-PTR TO ADDRESS OF DATA-TYPE.
       SET ERR-PTR  TO ADDRESS OF ERR-TYPE.

       MOVE  1 TO REGION.
       MOVE 66 TO TTYPE.
       MOVE  0 TO PRIORITY.
       MOVE  0 TO LOAD-BALANCE.

       MOVE 230               TO DATA-LEN OF DATA-TYPE.
       MOVE SNDTO-SRVR(1:230) TO DATA-DAT OF DATA-TYPE.

       EXEC CICS LINK PROGRAM('AEGTMRUN') COMMAREA(COMMAREAD)
               LENGTH(801) END-EXEC.

       MOVE DATA-DAT(1:670) TO STKLVL-O.
       MOVE BLWSTK OF STKLVL-O TO SLRX-BLWSTK.
       IF BLWSTK OF STKLVL-O = 0 MOVE "00" TO SLRX-BLWSTK.
       MOVE SLRX-BLWSTK TO BLWSTK OF DSPREC51-O.
HANDLE-STKLVL-EXIT.
     EXIT.

HANDLE-ORDSTS.
       MOVE PRM-WH1 TO ORDSTS-WAREHOUSE OF DISPLAY-ORDSTSIN.
            MOVE ODID  OF DISPLAY-ORDSTSIN TO DID      OF ORDSTS-I
       IF OCID OF DISPLAY-ORDSTSIN EQUAL "    " MOVE "0000" TO
          OCID OF DISPLAY-ORDSTSIN
       END-IF
       IF OCID OF DISPLAY-ORDSTSIN NOT EQUAL "0000"
          MOVE "C" TO ORDSTS-TYPE
       ELSE
          IF CLAST OF DISPLAY-ORDSTSIN NOT = SPACES
             MOVE "L" TO ORDSTS-TYPE
          ELSE
             MOVE "31" TO ORDSTS-FMT-NUM
             MOVE "31" TO HIDEME OF DSPREC31
             GO TO HANDLE-ORDSTS-EXIT
          END-IF
       END-IF
           MOVE OCID  OF DISPLAY-ORDSTSIN TO CID      OF ORDSTS-I
           MOVE CLAST OF DISPLAY-ORDSTSIN TO CLAST    OF ORDSTS-I
           MOVE ORDSTS-WAREHOUSE          TO WID      OF ORDSTS-I


       MOVE "O"        TO TXNTYPE  OF SNDTO-SRVR.
       MOVE PRM-JOBNAM TO SNDJOBNM OF SNDTO-SRVR.
       MOVE ORDSTS-I   TO TXNDATA  OF SNDTO-SRVR.

       SET DATA-PTR TO ADDRESS OF DATA-TYPE.
       SET ERR-PTR  TO ADDRESS OF ERR-TYPE.

       MOVE  1 TO REGION.
       MOVE 44 TO TTYPE.
       MOVE  0 TO PRIORITY.
       MOVE  0 TO LOAD-BALANCE.

       MOVE 230               TO DATA-LEN OF DATA-TYPE.
       MOVE SNDTO-SRVR(1:230) TO DATA-DAT OF DATA-TYPE.

       EXEC CICS LINK PROGRAM('AEGTMRUN') COMMAREA(COMMAREAD)
               LENGTH(801) END-EXEC.

       MOVE DATA-DAT(1:670) TO ORDSTS-O.

           MOVE ORDSTS-FMT-NUM1 OF ORDSTS-O TO
                ORDSTS-FMT-NUM OF ORDSTS-OUTDSP2
           IF ORDSTS-FMT-NUM1 OF ORDSTS-O = "62"
              GO TO HANDLE-ORDSTS-EXIT
           END-IF.
           MOVE "b" TO TORD
           CALL "getdttmstr" using OENTTOD OF ORDSTS-O,
                   ORDDTTIM , BY VALUE TORD


           MOVE ORDDT              TO OENTDT OF ORDSTS-OUTDSP2
           MOVE ORDTM              TO OENTTM OF ORDSTS-OUTDSP2
           MOVE CLAST  OF ORDSTS-O TO CLAST  OF ORDSTS-OUTDSP2
           MOVE CFIRST OF ORDSTS-O TO CFIRST OF ORDSTS-OUTDSP2
           MOVE CINIT  OF ORDSTS-O TO CINIT  OF ORDSTS-OUTDSP2
           MOVE OID    OF ORDSTS-O TO OID    OF ORDSTS-OUTDSP2
           MOVE OCID   OF ORDSTS-O TO OCIDN  OF ORDSTS-OUTDSP2
           MOVE CBAL   OF ORDSTS-O TO CBAL   OF ORDSTS-OUTDSP2
           MOVE OCARID OF ORDSTS-O TO OCARID OF ORDSTS-OUTDSP2
           ADD 30 TO OLINENBR GIVING OLINENBR
         IF OLINENBR < 35 MOVE 35 TO NUMBER-OF-ITEMS OF FORMAT-ID
         ELSE MOVE OLINENBR TO NUMBER-OF-ITEMS OF FORMAT-ID
         END-IF
           MOVE FMT-NUM TO ORDSTS-FMT-NUM OF ORDSTS-OUTDSP2
           SUBTRACT 30 FROM OLINENBR GIVING OLINENBR
           PERFORM VARYING I FROM
                1 BY 1 UNTIL I > OLINENBR
           MOVE OLINEINFO(I) TO OLINEINF
               MOVE DO-OLSPWH1 TO DO-OLSPWH2
               MOVE DO-OLIID1 TO DO-OLIID2
               MOVE DO-OLQTY1 TO DO-OLQTY2
               MOVE DO-OLAMNT1 TO DO-OLAMNT2


           MOVE "d" TO TORD
           CALL "getdttmstr" using DO-OLDLVD1,
                   OLDLVDT  , BY VALUE TORD

               MOVE OLDLVDT    TO DO-OLDLVD2
           MOVE DOLINE TO DO-LINE(I)
           END-PERFORM.
           PERFORM VARYING J FROM
                I BY 1 UNTIL J > 5
           MOVE DOLINE-NULL TO DO-LINE(J)
           END-PERFORM.
HANDLE-ORDSTS-EXIT.

SET-UP-ROUTINE.
       MOVE PRM-WH1 TO PRM-WH.
       MOVE PRM-DIST1 TO PRM-DIST.
***OPEN FILES ***
       OPEN I-O    DISPLAY-FILE.
*** Following lines MUST wait for display-file to be opened.
       MOVE "99" TO FMT-NUM.
       ACCEPT DATEINT     FROM DATE.
       ACCEPT CURTIME FROM TIME.
       MOVE YY TO TPCYEAR.
       MOVE MMDD TO TPCMD.
       IF YY < 70 THEN
            MOVE 20 TO TPCCEN.
       MOVE  TPCDATE TO OENTDT OF DSPREC00-O.
       MOVE SPACE-LINE TO INPUT-LINE(16).
       MOVE  PRM-WH  TO CWID OF DSPREC00-O.

END-OF-JOB.
       STOP RUN.
```

DLYFOREVER: Delay Forever

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>


void DLYFOREVER()
{
  char   cmd[16];


  memcpy(cmd,"DLYJOB DLY(9999)",16);
  cmd[16] = '\0';

  if ( system(cmd) )
  {
  }

}
```

ATPCCDSPF: Screen Definition File

```
    A*%%TS  SD  19920929  220345  PPCOC       REL-V2R2M0  5738-PW1
   *A*  This is the screen definition file for ALL screens used
   *A*    during execution of the TPC-C V2.0 Specification.
   *A*
    A*%%EC
    A                                      DSPSIZ(24 80 *DS3)
    A                                      REF(*LIBL/AREFFIL)
    A                                      CHGINPDFT(UL FE)
    A                                      CA01
    A                                      CA02
    A                                      CA03
    A                                      CA04
    A                                      CA05
    A                                      CA06
    A*DSPREC99 IS INITIAL SIGNON SCREEN - SETUP FOR NEWORD
    A*INPUT OR MENU SELECTION -ONLY TIME MENU IS WRITTEN-
    A          R DSPREC99
    A*%%TS  SD  19920830  082531  PPCOC       REL-V2R2M0  5738-PW1
    A                                   1 36'New Order'
    A                                     DSPATR(HI)
    A                                   2  1'Warehouse:'
    A            CWID      R        O  2 12REFFLD(CSRCD/CWID CSTMR)
    A                                   2 19'District:'
    A            CDID      R        I  2 29REFFLD(DSPDID)
                                          CHECK(ME RZ)
    A                                   2 40'Date:'
    A            OENTDT    R        O  2 46REFFLD(ORRCD/OENTDT ORDERS)
    A                                     EDTWRD('  -  -    ')
    A                                   2 60'Time:'
    A                                   2 66TIME
    A                                   3  1'Customer:'
    A            CID            4D  I  3 12
    A                                     CHECK(RZ ME)
    A                                   3 19'Name:'
    A                                   3 44'Credit:'
    A                                   3 57'%Disc:'
    A                                   4  1'Order Number:'
    A                                   4 25'Number of Lines:'
    A                                   4 52'W_tax:'
    A                                   4 67'D_tax:'
    A                                   6  2'Supp-W'
    A                                   6 10'Item Id'
    A                                   6 21'Qty'
    A                                   6 31'Item'
    A                                   6 36'Name'
    A                                   6 50'Stock'
    A                                   6 57'B/G'
    A                                   6 63'Price'
    A                                   6 73'Amount'
    A            OLSPWH1        4D  I  7  3
    A                                     CHECK(RZ ME)
    A            OLIID1         6D  I  7 11
    A                                     CHECK(RZ ME)
    A            OLQTY1    R         I  7 21REFFLD(DSPQTY)
    A                                     CHECK(RZ ME)
    A            OLSPWH2        4D  I  8  3
    A                                     CHECK(RZ)
    A            OLIID2         6D  I  8 11
    A                                     CHECK(RZ)
    A            OLQTY2    R         I  8 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH3        4D  I  9  3
    A                                     CHECK(RZ)
    A            OLIID3         6D  I  9 11
    A                                     CHECK(RZ)
    A            OLQTY3    R         I  9 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH4        4D  I 10  3
    A                                     CHECK(RZ)
    A            OLIID4         6D  I 10 11
    A                                     CHECK(RZ)
    A            OLQTY4    R         I 10 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH5        4D  I 11  3
    A                                     CHECK(RZ)
    A            OLIID5         6D  I 11 11
    A                                     CHECK(RZ)
    A            OLQTY5    R         I 11 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH6        4D  I 12  3
    A                                     CHECK(RZ)
    A            OLIID6         6D  I 12 11
    A                                     CHECK(RZ)
    A            OLQTY6    R         I 12 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH7        4D  I 13  3
    A                                     CHECK(RZ)
    A            OLIID7         6D  I 13 11
    A                                     CHECK(RZ)
    A            OLQTY7    R         I 13 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH8        4D  I 14  3
    A                                     CHECK(RZ)
    A            OLIID8         6D  I 14 11
    A                                     CHECK(RZ)
    A            OLQTY8    R         I 14 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH9        4D  I 15  3
    A                                     CHECK(RZ)
    A            OLIID9         6D  I 15 11
    A                                     CHECK(RZ)
    A            OLQTY9    R         I 15 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH10       4D  I 16  3
    A                                     CHECK(RZ)
    A            OLIID10        6D  I 16 11
    A                                     CHECK(RZ)
    A            OLQTY10   R         I 16 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH11       4D  I 17  3
    A                                     CHECK(RZ)
    A            OLIID11        6D  I 17 11
    A                                     CHECK(RZ)
    A            OLQTY11   R         I 17 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH12       4D  I 18  3
    A                                     CHECK(RZ)
    A            OLIID12        6D  I 18 11
    A                                     CHECK(RZ)
    A            OLQTY12   R         I 18 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH13       4D  I 19  3
    A                                     CHECK(RZ)
    A            OLIID13        6D  I 19 11
    A                                     CHECK(RZ)
    A            OLQTY13   R         I 19 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH14       4D  I 20  3
    A                                     CHECK(RZ)
    A            OLIID14        6D  I 20 11
    A                                     CHECK(RZ)
    A            OLQTY14   R         I 20 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH15       4D  I 21  3
    A                                     CHECK(RZ)
    A            OLIID15        6D  I 21 11
    A                                     CHECK(RZ)
    A            OLQTY15   R         I 21 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A                                  22  2'EXECUTION STATUS:'
    A                                  22 54'Total:'
    A                                  23  2'PF1-          PF2-         PF3-    -
    A                                     PF4-         PF5-         PF6-'
    A                                  24  2'New Order    Payment      EXIT    -
    A                                       Delivery     Order Status Stoc-
    A                                     k Level'
    A          R DSPREC00
    A*%%TS  SD  19920830  082531  PPCOC       REL-V2R2M0  5738-PW1
    A                                     CLRL(22)
    A                                   1 36'New Order'
    A                                     DSPATR(HI)
    A                                   2  1'Warehouse:'
    A            CWID      R        O  2 12REFFLD(CSRCD/CWID CSTMR)
    A                                   2 19'District:'
    A            CDID      R        I  2 29REFFLD(DSPDID)
                                          CHECK(ME RZ)
    A                                   2 40'Date:'
    A            OENTDT    R        O  2 46REFFLD(ORRCD/OENTDT ORDERS)
    A                                     EDTWRD('  -  -    ')
    A                                   2 60'Time:'
    A                                   2 66TIME
    A                                   3  1'Customer:'
    A            CID            4D  I  3 12
    A                                     CHECK(RZ ME)
    A                                   3 19'Name:'
    A                                   3 44'Credit:'
    A                                   3 57'%Disc:'
    A                                   4  1'Order Number:'
    A                                   4 25'Number of Lines:'
    A                                   4 52'W_tax:'
    A                                   4 67'D_tax:'
    A                                   6  2'Supp-W'
    A                                   6 10'Item Id'
    A                                   6 21'Qty'
    A                                   6 31'Item'
    A                                   6 36'Name'
    A                                   6 50'Stock'
    A                                   6 57'B/G'
    A                                   6 63'Price'
    A                                   6 73'Amount'
    A            OLSPWH1        4D  I  7  3
    A                                     CHECK(RZ ME)
    A            OLIID1         6D  I  7 11
    A                                     CHECK(RZ ME)
    A            OLQTY1    R         I  7 21REFFLD(DSPQTY)
    A                                     CHECK(RZ ME)
    A            OLSPWH2        4D  I  8  3
    A                                     CHECK(RZ)
    A            OLIID2         6D  I  8 11
    A                                     CHECK(RZ)
    A            OLQTY2    R         I  8 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH3        4D  I  9  3
    A                                     CHECK(RZ)
    A            OLIID3         6D  I  9 11
    A                                     CHECK(RZ)
    A            OLQTY3    R         I  9 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH4        4D  I 10  3
    A                                     CHECK(RZ)
    A            OLIID4         6D  I 10 11
    A                                     CHECK(RZ)
    A            OLQTY4    R         I 10 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH5        4D  I 11  3
    A                                     CHECK(RZ)
    A            OLIID5         6D  I 11 11
    A                                     CHECK(RZ)
    A            OLQTY5    R         I 11 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH6        4D  I 12  3
    A                                     CHECK(RZ)
    A            OLIID6         6D  I 12 11
    A                                     CHECK(RZ)
    A            OLQTY6    R         I 12 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH7        4D  I 13  3
    A                                     CHECK(RZ)
    A            OLIID7         6D  I 13 11
    A                                     CHECK(RZ)
    A            OLQTY7    R         I 13 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH8        4D  I 14  3
    A                                     CHECK(RZ)
    A            OLIID8         6D  I 14 11
    A                                     CHECK(RZ)
    A            OLQTY8    R         I 14 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH9        4D  I 15  3
    A                                     CHECK(RZ)
    A            OLIID9         6D  I 15 11
    A                                     CHECK(RZ)
    A            OLQTY9    R         I 15 21REFFLD(DSPQTY)
    A                                     CHECK(RZ)
    A            OLSPWH10       4D  I 16  3
    A                                     CHECK(RZ)
    A            OLIID10        6D  I 16 11
```

```
A                                      CHECK(RZ)
A            OLQTY10   R        I 16 21REFFLD(DSPQTY)
A                                      CHECK(RZ)
A            OLSPWH11        4D  I 17  3
A                                      CHECK(RZ)
A            OLIID11         6D  I 17 11
A                                      CHECK(RZ)
A            OLQTY11   R        I 17 21REFFLD(DSPQTY)
A                                      CHECK(RZ)
A            OLSPWH12        4D  I 18  3
A                                      CHECK(RZ)
A            OLIID12         6D  I 18 11
A                                      CHECK(RZ)
A            OLQTY12   R        I 18 21REFFLD(DSPQTY)
A                                      CHECK(RZ)
A            OLSPWH13        4D  I 19  3
A                                      CHECK(RZ)
A            OLIID13         6D  I 19 11
A                                      CHECK(RZ)
A            OLQTY13   R        I 19 21REFFLD(DSPQTY)
A                                      CHECK(RZ)
A            OLSPWH14        4D  I 20  3
A                                      CHECK(RZ)
A            OLIID14         6D  I 20 11
A                                      CHECK(RZ)
A            OLQTY14   R        I 20 21REFFLD(DSPQTY)
A                                      CHECK(RZ)
A            OLSPWH15        4D  I 21  3
A                                      CHECK(RZ)
A            OLIID15         6D  I 21 11
A                                      CHECK(RZ)
A            OLQTY15   R        I 21 21REFFLD(DSPQTY)
A                                      CHECK(RZ)
A                                    22  2'EXECUTION STATUS:'
A                                    22 54'Total:'
A          R DSPRECRB
A*%%TS  SD  19920819  111811  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME          2   H
A            CID       R        O  3 12REFFLD(CSRCD/CID   CSTMR)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            OID       R        O  4 15REFFLD(DSPOID)
A                                    22 20'ITEM NUMBER IS NOT VALID'
A                                       DSPATR(HI)
A          R DSPRECRC
A*%%TS  SD  19920819  111811  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME          2   H
A            CID       R        O  3 12REFFLD(CSRCD/CID   CSTMR)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            OID       R        O  4 15REFFLD(DSPOID)
A                                    22 20'SUPPLY WRHS IS NOT VALID'
A                                       DSPATR(HI)
A          R DSPRECRD
A*%%TS  SD  19920819  111811  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME          2   H
A            CID       R        O  3 12REFFLD(CSRCD/CID   CSTMR)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            OID       R        O  4 15REFFLD(DSPOID)
A                                    22 20'ITEM QTY IS NOT VALID'
A                                       DSPATR(HI)
A          R DSPREC02
A*%%TS  SD  19920929  220345  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME          2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL  CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT            4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX            4Y 2O  4 59EDTCDE(1)
A            DTAX            4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1           1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC03
A*%%TS  SD  19920929  220345  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME          2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT            4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX            4Y 2O  4 59EDTCDE(1)
A            DTAX            4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1           1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
A            BORG2           1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC04
A*%%TS  SD  19920929  220345  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME          2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT            4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX            4Y 2O  4 59EDTCDE(1)
A            DTAX            4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1           1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
A            BORG2           1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME3    R        O  9 25REFFLD(ITRCD/INAME ITEM)
A            STQTY3    R        O  9 51REFFLD(QTY3)
A            BORG3           1A  O  9 58
A            IPRICE3   R        O  9 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT3   R        O  9 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC05
A*%%TS  SD  19920929  220345  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME          2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT            4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX            4Y 2O  4 59EDTCDE(1)
A            DTAX            4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1           1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
A            BORG2           1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME3    R        O  9 25REFFLD(ITRCD/INAME ITEM)
A            STQTY3    R        O  9 51REFFLD(QTY3)
A            BORG3           1A  O  9 58
A            IPRICE3   R        O  9 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT3   R        O  9 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME4    R        O 10 25REFFLD(ITRCD/INAME ITEM)
A            STQTY4    R        O 10 51REFFLD(QTY3)
A            BORG4           1A  O 10 58
A            IPRICE4   R        O 10 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT4   R        O 10 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC06
A*%%TS  SD  19920929  220345  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME          2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT            4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX            4Y 2O  4 59EDTCDE(1)
A            DTAX            4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1           1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
A            BORG2           1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME3    R        O  9 25REFFLD(ITRCD/INAME ITEM)
A            STQTY3    R        O  9 51REFFLD(QTY3)
A            BORG3           1A  O  9 58
A            IPRICE3   R        O  9 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT3   R        O  9 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME4    R        O 10 25REFFLD(ITRCD/INAME ITEM)
A            STQTY4    R        O 10 51REFFLD(QTY3)
A            BORG4           1A  O 10 58
A            IPRICE4   R        O 10 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT4   R        O 10 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC07
A*%%TS  SD  19920819  081833  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME          2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT            4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX            4Y 2O  4 59EDTCDE(1)
A            DTAX            4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1           1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
A            BORG2           1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME3    R        O  9 25REFFLD(ITRCD/INAME ITEM)
A            STQTY3    R        O  9 51REFFLD(QTY3)
A            BORG3           1A  O  9 58
A            IPRICE3   R        O  9 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT3   R        O  9 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME4    R        O 10 25REFFLD(ITRCD/INAME ITEM)
A            STQTY4    R        O 10 51REFFLD(QTY3)
A            BORG4           1A  O 10 58
A            IPRICE4   R        O 10 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT4   R        O 10 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME5    R        O 11 25REFFLD(ITRCD/INAME ITEM)
A            STQTY5    R        O 11 51REFFLD(QTY3)
A            BORG5           1A  O 11 58
A            IPRICE5   R        O 11 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT5   R        O 11 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC08
A*%%TS  SD  19920819  081833  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME          2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT            4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX            4Y 2O  4 59EDTCDE(1)
A            DTAX            4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1           1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
```

```
A            BORG2          1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME3    R        O  9 25REFFLD(ITRCD/INAME ITEM)
A            STQTY3    R        O  9 51REFFLD(QTY3)
A            BORG3          1A  O  9 58
A            IPRICE3   R        O  9 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT3   R        O  9 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME4    R        O 10 25REFFLD(ITRCD/INAME ITEM)
A            STQTY4    R        O 10 51REFFLD(QTY3)
A            BORG4          1A  O 10 58
A            IPRICE4   R        O 10 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT4   R        O 10 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME5    R        O 11 25REFFLD(ITRCD/INAME ITEM)
A            STQTY5    R        O 11 51REFFLD(QTY3)
A            BORG5          1A  O 11 58
A            IPRICE5   R        O 11 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT5   R        O 11 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME6    R        O 12 25REFFLD(ITRCD/INAME ITEM)
A            STQTY6    R        O 12 51REFFLD(QTY3)
A            BORG6          1A  O 12 58
A            IPRICE6   R        O 12 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT6   R        O 12 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME7    R        O 13 25REFFLD(ITRCD/INAME ITEM)
A            STQTY7    R        O 13 51REFFLD(QTY3)
A            BORG7          1A  O 13 58
A            IPRICE7   R        O 13 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT7   R        O 13 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC09
A*%%TS  SD  19920819  081833  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME         2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT           4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX           4Y 2O  4 59EDTCDE(1)
A            DTAX           4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1          1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
A            BORG2          1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME3    R        O  9 25REFFLD(ITRCD/INAME ITEM)
A            STQTY3    R        O  9 51REFFLD(QTY3)
A            BORG3          1A  O  9 58
A            IPRICE3   R        O  9 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT3   R        O  9 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME4    R        O 10 25REFFLD(ITRCD/INAME ITEM)
A            STQTY4    R        O 10 51REFFLD(QTY3)
A            BORG4          1A  O 10 58
A            IPRICE4   R        O 10 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT4   R        O 10 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME5    R        O 11 25REFFLD(ITRCD/INAME ITEM)
A            STQTY5    R        O 11 51REFFLD(QTY3)
A            BORG5          1A  O 11 58
A            IPRICE5   R        O 11 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT5   R        O 11 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME6    R        O 12 25REFFLD(ITRCD/INAME ITEM)
A            STQTY6    R        O 12 51REFFLD(QTY3)
A            BORG6          1A  O 12 58
A            IPRICE6   R        O 12 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT6   R        O 12 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME7    R        O 13 25REFFLD(ITRCD/INAME ITEM)
A            STQTY7    R        O 13 51REFFLD(QTY3)
A            BORG7          1A  O 13 58
A            IPRICE7   R        O 13 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT7   R        O 13 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME8    R        O 14 25REFFLD(ITRCD/INAME ITEM)
A            STQTY8    R        O 14 51REFFLD(QTY3)
A            BORG8          1A  O 14 58
A            IPRICE8   R        O 14 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT8   R        O 14 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC10
A*%%TS  SD  19920819  081833  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME         2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT           4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX           4Y 2O  4 59EDTCDE(1)
A            DTAX           4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1          1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
A            BORG2          1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME3    R        O  9 25REFFLD(ITRCD/INAME ITEM)
A            STQTY3    R        O  9 51REFFLD(QTY3)
A            BORG3          1A  O  9 58
A            IPRICE3   R        O  9 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT3   R        O  9 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME4    R        O 10 25REFFLD(ITRCD/INAME ITEM)
A            STQTY4    R        O 10 51REFFLD(QTY3)
A            BORG4          1A  O 10 58
A            IPRICE4   R        O 10 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT4   R        O 10 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME5    R        O 11 25REFFLD(ITRCD/INAME ITEM)
A            STQTY5    R        O 11 51REFFLD(QTY3)
A            BORG5          1A  O 11 58
A            IPRICE5   R        O 11 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT5   R        O 11 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME6    R        O 12 25REFFLD(ITRCD/INAME ITEM)
A            STQTY6    R        O 12 51REFFLD(QTY3)
A            BORG6          1A  O 12 58
A            IPRICE6   R        O 12 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT6   R        O 12 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME7    R        O 13 25REFFLD(ITRCD/INAME ITEM)
A            STQTY7    R        O 13 51REFFLD(QTY3)
A            BORG7          1A  O 13 58
A            IPRICE7   R        O 13 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT7   R        O 13 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME8    R        O 14 25REFFLD(ITRCD/INAME ITEM)
A            STQTY8    R        O 14 51REFFLD(QTY3)
A            BORG8          1A  O 14 58
A            IPRICE8   R        O 14 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT8   R        O 14 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME9    R        O 15 25REFFLD(ITRCD/INAME ITEM)
A            STQTY9    R        O 15 51REFFLD(QTY3)
A            BORG9          1A  O 15 58
A            IPRICE9   R        O 15 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT9   R        O 15 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC11
A*%%TS  SD  19920819  081833  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME         2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT           4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX           4Y 2O  4 59EDTCDE(1)
A            DTAX           4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1          1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
A            BORG2          1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME3    R        O  9 25REFFLD(ITRCD/INAME ITEM)
A            STQTY3    R        O  9 51REFFLD(QTY3)
A            BORG3          1A  O  9 58
A            IPRICE3   R        O  9 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT3   R        O  9 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME4    R        O 10 25REFFLD(ITRCD/INAME ITEM)
A            STQTY4    R        O 10 51REFFLD(QTY3)
A            BORG4          1A  O 10 58
A            IPRICE4   R        O 10 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT4   R        O 10 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME5    R        O 11 25REFFLD(ITRCD/INAME ITEM)
A            STQTY5    R        O 11 51REFFLD(QTY3)
A            BORG5          1A  O 11 58
A            IPRICE5   R        O 11 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT5   R        O 11 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME6    R        O 12 25REFFLD(ITRCD/INAME ITEM)
A            STQTY6    R        O 12 51REFFLD(QTY3)
A            BORG6          1A  O 12 58
A            IPRICE6   R        O 12 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT6   R        O 12 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME7    R        O 13 25REFFLD(ITRCD/INAME ITEM)
A            STQTY7    R        O 13 51REFFLD(QTY3)
A            BORG7          1A  O 13 58
A            IPRICE7   R        O 13 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT7   R        O 13 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME8    R        O 14 25REFFLD(ITRCD/INAME ITEM)
A            STQTY8    R        O 14 51REFFLD(QTY3)
A            BORG8          1A  O 14 58
A            IPRICE8   R        O 14 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT8   R        O 14 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME9    R        O 15 25REFFLD(ITRCD/INAME ITEM)
A            STQTY9    R        O 15 51REFFLD(QTY3)
A            BORG9          1A  O 15 58
A            IPRICE9   R        O 15 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT9   R        O 15 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME10   R        O 16 25REFFLD(ITRCD/INAME ITEM)
A            STQTY10   R        O 16 51REFFLD(QTY3)
A            BORG10         1A  O 16 58
A            IPRICE10  R        O 16 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT910 R        O 16 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC12
A*%%TS  SD  19920819  081833  PPCOC       REL-V2R2M0  5738-PW1
A                                      CLRL(*NO)
A            HIDEME         2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT           4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX           4Y 2O  4 59EDTCDE(1)
A            DTAX           4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1          1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
A            BORG2          1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME3    R        O  9 25REFFLD(ITRCD/INAME ITEM)
A            STQTY3    R        O  9 51REFFLD(QTY3)
A            BORG3          1A  O  9 58
A            IPRICE3   R        O  9 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT3   R        O  9 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME4    R        O 10 25REFFLD(ITRCD/INAME ITEM)
A            STQTY4    R        O 10 51REFFLD(QTY3)
A            BORG4          1A  O 10 58
A            IPRICE4   R        O 10 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT4   R        O 10 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME5    R        O 11 25REFFLD(ITRCD/INAME ITEM)
A            STQTY5    R        O 11 51REFFLD(QTY3)
A            BORG5          1A  O 11 58
A            IPRICE5   R        O 11 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT5   R        O 11 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME6    R        O 12 25REFFLD(ITRCD/INAME ITEM)
A            STQTY6    R        O 12 51REFFLD(QTY3)
A            BORG6          1A  O 12 58
A            IPRICE6   R        O 12 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT6   R        O 12 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME7    R        O 13 25REFFLD(ITRCD/INAME ITEM)
A            STQTY7    R        O 13 51REFFLD(QTY3)
A            BORG7          1A  O 13 58
A            IPRICE7   R        O 13 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT7   R        O 13 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME8    R        O 14 25REFFLD(ITRCD/INAME ITEM)
A            STQTY8    R        O 14 51REFFLD(QTY3)
A            BORG8          1A  O 14 58
A            IPRICE8   R        O 14 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT8   R        O 14 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME9    R        O 15 25REFFLD(ITRCD/INAME ITEM)
A            STQTY9    R        O 15 51REFFLD(QTY3)
A            BORG9          1A  O 15 58
A            IPRICE9   R        O 15 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT9   R        O 15 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME10   R        O 16 25REFFLD(ITRCD/INAME ITEM)
A            STQTY10   R        O 16 51REFFLD(QTY3)
A            BORG10         1A  O 16 58
A            IPRICE10  R        O 16 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT10  R        O 16 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME11   R        O 17 25REFFLD(ITRCD/INAME ITEM)
A            STQTY11   R        O 17 51REFFLD(QTY3)
A            BORG11         1A  O 17 58
```

```
A            IPRICE11  R        O 17 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT11  R        O 17 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC13
A*%%TS  SD  19920819  081833  PPCOC       REL-V2R2M0  5738-PW1
A                                        CLRL(*NO)
A            HIDEME          2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT            4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX            4Y 2O  4 59EDTCDE(1)
A            DTAX            4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1          1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
A            BORG2          1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME3    R        O  9 25REFFLD(ITRCD/INAME ITEM)
A            STQTY3    R        O  9 51REFFLD(QTY3)
A            BORG3          1A  O  9 58
A            IPRICE3   R        O  9 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT3   R        O  9 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME4    R        O 10 25REFFLD(ITRCD/INAME ITEM)
A            STQTY4    R        O 10 51REFFLD(QTY3)
A            BORG4          1A  O 10 58
A            IPRICE4   R        O 10 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT4   R        O 10 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME5    R        O 11 25REFFLD(ITRCD/INAME ITEM)
A            STQTY5    R        O 11 51REFFLD(QTY3)
A            BORG5          1A  O 11 58
A            IPRICE5   R        O 11 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT5   R        O 11 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME6    R        O 12 25REFFLD(ITRCD/INAME ITEM)
A            STQTY6    R        O 12 51REFFLD(QTY3)
A            BORG6          1A  O 12 58
A            IPRICE6   R        O 12 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT6   R        O 12 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME7    R        O 13 25REFFLD(ITRCD/INAME ITEM)
A            STQTY7    R        O 13 51REFFLD(QTY3)
A            BORG7          1A  O 13 58
A            IPRICE7   R        O 13 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT7   R        O 13 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME8    R        O 14 25REFFLD(ITRCD/INAME ITEM)
A            STQTY8    R        O 14 51REFFLD(QTY3)
A            BORG8          1A  O 14 58
A            IPRICE8   R        O 14 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT8   R        O 14 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME9    R        O 15 25REFFLD(ITRCD/INAME ITEM)
A            STQTY9    R        O 15 51REFFLD(QTY3)
A            BORG9          1A  O 15 58
A            IPRICE9   R        O 15 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT9   R        O 15 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME10   R        O 16 25REFFLD(ITRCD/INAME ITEM)
A            STQTY10   R        O 16 51REFFLD(QTY3)
A            BORG10         1A  O 16 58
A            IPRICE10  R        O 16 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT10  R        O 16 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME11   R        O 17 25REFFLD(ITRCD/INAME ITEM)
A            STQTY11   R        O 17 51REFFLD(QTY3)
A            BORG11         1A  O 17 58
A            IPRICE11  R        O 17 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT11  R        O 17 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME12   R        O 18 25REFFLD(ITRCD/INAME ITEM)
A            STQTY12   R        O 18 51REFFLD(QTY3)
A            BORG12         1A  O 18 58
A            IPRICE12  R        O 18 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT12  R        O 18 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC14
A*%%TS  SD  19920819  081833  PPCOC       REL-V2R2M0  5738-PW1
A                                        CLRL(*NO)
A            HIDEME          2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT            4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX            4Y 2O  4 59EDTCDE(1)
A            DTAX            4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1          1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
A            BORG2          1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME3    R        O  9 25REFFLD(ITRCD/INAME ITEM)
A            STQTY3    R        O  9 51REFFLD(QTY3)
A            BORG3          1A  O  9 58
A            IPRICE3   R        O  9 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT3   R        O  9 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME4    R        O 10 25REFFLD(ITRCD/INAME ITEM)
A            STQTY4    R        O 10 51REFFLD(QTY3)
A            BORG4          1A  O 10 58
A            IPRICE4   R        O 10 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT4   R        O 10 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME5    R        O 11 25REFFLD(ITRCD/INAME ITEM)
A            STQTY5    R        O 11 51REFFLD(QTY3)
A            BORG5          1A  O 11 58
A            IPRICE5   R        O 11 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT5   R        O 11 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME6    R        O 12 25REFFLD(ITRCD/INAME ITEM)
A            STQTY6    R        O 12 51REFFLD(QTY3)
A            BORG6          1A  O 12 58
A            IPRICE6   R        O 12 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT6   R        O 12 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME7    R        O 13 25REFFLD(ITRCD/INAME ITEM)
A            STQTY7    R        O 13 51REFFLD(QTY3)
A            BORG7          1A  O 13 58
A            IPRICE7   R        O 13 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT7   R        O 13 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME8    R        O 14 25REFFLD(ITRCD/INAME ITEM)
A            STQTY8    R        O 14 51REFFLD(QTY3)
A            BORG8          1A  O 14 58
A            IPRICE8   R        O 14 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT8   R        O 14 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME9    R        O 15 25REFFLD(ITRCD/INAME ITEM)
A            STQTY9    R        O 15 51REFFLD(QTY3)
A            BORG9          1A  O 15 58
A            IPRICE9   R        O 15 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT9   R        O 15 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME10   R        O 16 25REFFLD(ITRCD/INAME ITEM)
A            STQTY10   R        O 16 51REFFLD(QTY3)
A            BORG10         1A  O 16 58
A            IPRICE10  R        O 16 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT10  R        O 16 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME11   R        O 17 25REFFLD(ITRCD/INAME ITEM)
A            STQTY11   R        O 17 51REFFLD(QTY3)
A            BORG11         1A  O 17 58
A            IPRICE11  R        O 17 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT11  R        O 17 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME12   R        O 18 25REFFLD(ITRCD/INAME ITEM)
A            STQTY12   R        O 18 51REFFLD(QTY3)
A            BORG12         1A  O 18 58
A            IPRICE12  R        O 18 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT12  R        O 18 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME13   R        O 19 25REFFLD(ITRCD/INAME ITEM)
A            STQTY13   R        O 19 51REFFLD(QTY3)
A            BORG13         1A  O 19 58
A            IPRICE13  R        O 19 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT13  R        O 19 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC15
A*%%TS  SD  19920819  081833  PPCOC       REL-V2R2M0  5738-PW1
A                                        CLRL(*NO)
A            HIDEME          2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT            4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX            4Y 2O  4 59EDTCDE(1)
A            DTAX            4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1          1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
A            BORG2          1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME3    R        O  9 25REFFLD(ITRCD/INAME ITEM)
A            STQTY3    R        O  9 51REFFLD(QTY3)
A            BORG3          1A  O  9 58
A            IPRICE3   R        O  9 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT3   R        O  9 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME4    R        O 10 25REFFLD(ITRCD/INAME ITEM)
A            STQTY4    R        O 10 51REFFLD(QTY3)
A            BORG4          1A  O 10 58
A            IPRICE4   R        O 10 62REFFLD(ITRCD/IPRICE ITEM)
A            STQTY10   R        O 16 51REFFLD(QTY3)
A            BORG10         1A  O 16 58
A            IPRICE10  R        O 16 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT10  R        O 16 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME11   R        O 17 25REFFLD(ITRCD/INAME ITEM)
A            STQTY11   R        O 17 51REFFLD(QTY3)
A            BORG11         1A  O 17 58
A            IPRICE11  R        O 17 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT11  R        O 17 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME12   R        O 18 25REFFLD(ITRCD/INAME ITEM)
A            STQTY12   R        O 18 51REFFLD(QTY3)
A            BORG12         1A  O 18 58
A            IPRICE12  R        O 18 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT12  R        O 18 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME13   R        O 19 25REFFLD(ITRCD/INAME ITEM)
A            STQTY13   R        O 19 51REFFLD(QTY3)
A            BORG13         1A  O 19 58
A            IPRICE13  R        O 19 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT13  R        O 19 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME14   R        O 20 25REFFLD(ITRCD/INAME ITEM)
A            STQTY14   R        O 20 51REFFLD(QTY3)
A            BORG14         1A  O 20 58
A            IPRICE14  R        O 20 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT14  R        O 20 71REFFLD(OLRCD/OLAMNT ORDLIN)
A          R DSPREC16
A*%%TS  SD  19920819  081833  PPCOC       REL-V2R2M0  5738-PW1
A                                        CLRL(*NO)
A            HIDEME          2   H
A            TOTAMT    R        O 22 61REFFLD(CSRCD/CBAL CSTMR)
A            OENTTM    R        O  2 66REFFLD(ORRCD/OENTTM ORDERS)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            CDCT            4Y 2O  3 64EDTCDE(1)
A            OID       R        O  4 15REFFLD(DSPOID)
A            OLINES    R        O  4 42REFFLD(DSPOLN)
A            WTAX            4Y 2O  4 59EDTCDE(1)
A            DTAX            4Y 2O  4 74EDTCDE(1)
A            INAME1    R        O  7 25REFFLD(ITRCD/INAME ITEM)
A            STQTY1    R        O  7 51REFFLD(QTY3)
A            BORG1          1A  O  7 58
A            IPRICE1   R        O  7 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT1   R        O  7 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME2    R        O  8 25REFFLD(ITRCD/INAME ITEM)
A            STQTY2    R        O  8 51REFFLD(QTY3)
A            BORG2          1A  O  8 58
A            IPRICE2   R        O  8 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT2   R        O  8 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME3    R        O  9 25REFFLD(ITRCD/INAME ITEM)
A            STQTY3    R        O  9 51REFFLD(QTY3)
A            BORG3          1A  O  9 58
A            IPRICE3   R        O  9 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT3   R        O  9 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME4    R        O 10 25REFFLD(ITRCD/INAME ITEM)
A            STQTY4    R        O 10 51REFFLD(QTY3)
A            BORG4          1A  O 10 58
A            IPRICE4   R        O 10 62REFFLD(ITRCD/IPRICE ITEM)
```

```
A            OLAMNT4   R        O 10 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME5    R        O 11 25REFFLD(ITRCD/INAME ITEM)
A            STQTY5    R        O 11 51REFFLD(QTY3)
A            BORG5          1A  O 11 58
A            IPRICE5   R        O 11 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT5   R        O 11 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME6    R        O 12 25REFFLD(ITRCD/INAME ITEM)
A            STQTY6    R        O 12 51REFFLD(QTY3)
A            BORG6          1A  O 12 58
A            IPRICE6   R        O 12 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT6   R        O 12 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME7    R        O 13 25REFFLD(ITRCD/INAME ITEM)
A            STQTY7    R        O 13 51REFFLD(QTY3)
A            BORG7          1A  O 13 58
A            IPRICE7   R        O 13 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT7   R        O 13 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME8    R        O 14 25REFFLD(ITRCD/INAME ITEM)
A            STQTY8    R        O 14 51REFFLD(QTY3)
A            BORG8          1A  O 14 58
A            IPRICE8   R        O 14 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT8   R        O 14 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME9    R        O 15 25REFFLD(ITRCD/INAME ITEM)
A            STQTY9    R        O 15 51REFFLD(QTY3)
A            BORG9          1A  O 15 58
A            IPRICE9   R        O 15 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT9   R        O 15 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME10   R        O 16 25REFFLD(ITRCD/INAME ITEM)
A            STQTY10   R        O 16 51REFFLD(QTY3)
A            BORG10         1A  O 16 58
A            IPRICE10  R        O 16 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT10  R        O 16 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME11   R        O 17 25REFFLD(ITRCD/INAME ITEM)
A            STQTY11   R        O 17 51REFFLD(QTY3)
A            BORG11         1A  O 17 58
A            IPRICE11  R        O 17 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT11  R        O 17 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME12   R        O 18 25REFFLD(ITRCD/INAME ITEM)
A            STQTY12   R        O 18 51REFFLD(QTY3)
A            BORG12         1A  O 18 58
A            IPRICE12  R        O 18 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT12  R        O 18 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME13   R        O 19 25REFFLD(ITRCD/INAME ITEM)
A            STQTY13   R        O 19 51REFFLD(QTY3)
A            BORG13         1A  O 19 58
A            IPRICE13  R        O 19 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT13  R        O 19 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME14   R        O 20 25REFFLD(ITRCD/INAME ITEM)
A            STQTY14   R        O 20 51REFFLD(QTY3)
A            BORG14         1A  O 20 58
A            IPRICE14  R        O 20 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT14  R        O 20 71REFFLD(OLRCD/OLAMNT ORDLIN)
A            INAME15   R        O 21 25REFFLD(ITRCD/INAME ITEM)
A            STQTY15   R        O 21 51REFFLD(QTY3)
A            BORG15         1A  O 21 58
A            IPRICE15  R        O 21 62REFFLD(ITRCD/IPRICE ITEM)
A            OLAMNT15  R        O 21 71REFFLD(OLRCD/OLAMNT ORDLIN)
*A*  Payment Screen
 A*  INITIAL INPUT SCREEN
 A          R DSPREC88
 A                                      CLRL(22)
 A                                    1 38'Payment'
 A                                    2  1'Date:'
 A                                    4  1'Warehouse:'
 A            WID       R        O  4 12REFFLD(WRRCD/WID WRHS)
 A                                    4 42'District:'
 A            DID       R        I  4 52REFFLD(DSPDID)
 A                                      CHECK(ME RZ)
 A                                    9  1'Customer:'
 A            CID            4D  I  9 11
 A                                      DSPATR(MDT)
 A                                      CHECK(RZ)
 A                                    9 17'Cust-Warehouse:'
 A            CWID           4D  I  9 33
 A                                      CHECK(RZ ME)
 A                                    9 39'Cust-District:'
 A            CDID      R        I  9 54REFFLD(DSPDID)
 A                                      CHECK(ME RZ)
 A                                   10  1'Name:'
 A            CLAST     R        I 10  9REFFLD(CSRCD/CLAST CSTMR)
 A                                   10 50'Since:'
 A                                   11 50'Credit:'
 A                                   12 50'%Disc:'
 A                                   13 50'Phone:'
 A                                   15  1'Amount Paid:'
 A            PAYMNT         6Y 2I 15 15TEXT('Amount')
 A                                      CHECK(RZ ME)
 A                                   15 37'New Cust Balance:'
 A                                   16  1'Credit Limit:'
 A                                   18  1'Cust-Data:'
 A* PAYMENT OUTPUT DISPLAY
 A          R DSPREC17
 A                                      CLRL(*NO)
 A            HIDEME         2   H
 A            HDATE     R        O  2  7REFFLD(HSRCD/HDATE HSTRY)
 A            HTIME     R        O  2 18REFFLD(HSRCD/HTIME HSTRY)
 A            WADDR1    R        O  5  1REFFLD(WRRCD/WADDR1 WRHS)
 A            DADDR1    R        O  5 42REFFLD(DSRCD/DADDR1 DSTRCT)
 A            WADDR2    R        O  6  1REFFLD(WRRCD/WADDR2 WRHS)
 A            DADDR2    R        O  6 42REFFLD(DSRCD/DADDR2 DSTRCT)
 A            WCITY     R        O  7  1REFFLD(WRRCD/WCITY WRHS)
 A            WSTATE    R        O  7 22REFFLD(WRRCD/WSTATE WRHS)
 A            WZIP      R        O  7 25REFFLD(WRRCD/WZIP   WRHS)
 A            DCITY     R        O  7 42REFFLD(DSRCD/DCITY  DSTRCT)
 A            DSTATE    R        O  7 62REFFLD(DSRCD/DSTATE DSTRCT)
 A            DZIP      R        O  7 66REFFLD(DSRCD/DZIP   DSTRCT)
 A            CID       R        O  9 11REFFLD(CSRCD/CID   CSTMR)
 A            CFIRST    R        O 10  9REFFLD(CSRCD/CFIRST CSTMR)
 A            CINIT     R        O 10 26REFFLD(CSRCD/CINIT CSTMR)
 A            CLAST     R        O 10 29REFFLD(CSRCD/CLAST CSTMR)
 A            CLDATE    R        O 10 58REFFLD(CSRCD/CLDATE CSTMR)
 A            CADDR1    R        O 11  9REFFLD(CSRCD/CADDR1 CSTMR)
 A            CCREDT    R        O 11 58REFFLD(CSRCD/CCREDT CSTMR)
 A            CADDR2    R        O 12  9REFFLD(CSRCD/CADDR2 CSTMR)
 A            CDCT           4  2O 12 58EDTCDE(1)
 A            CCITY     R        O 13  9REFFLD(CSRCD/CCITY CSTMR)
 A            CSTATE    R        O 13 30REFFLD(CSRCD/CSTATE CSTMR)
 A            CZIP      R        O 13 33REFFLD(CSRCD/CZIP CSTMR)
 A            CPHONE         19A  O 13 58
 A            CBAL      R        O 15 55REFFLD(CSRCD/CBAL CSTMR)
 A            PAYMNT         6Y 2O 15 15EDTCDE(J $)
 A            CCRDLM    R        O 16 15REFFLD(CSRCD/CCRDLM CSTMR)
A* The next 4 fields are Logical Version of CDATA via COBOL renames.
 A            CDAT1          50   O 18 12
 A            CDAT2          50   O 19 12
 A            CDAT3          50   O 20 12
 A            CDAT4          50   O 21 12
 A          R DSPREC18
 A                                      CLRL(*NO)
 A            HIDEME         2   H
 A                                    22 20'ENTER CID OR CLAST'
 A                                      DSPATR(HI)
 A* PAYMENT OUTPUT DISPLAY WITHOUT BAD CREDIT STUFF
 A          R DSPREC19
 A                                      CLRL(*NO)
 A            HIDEME         2   H
 A            HDATE     R        O  2  7REFFLD(HSRCD/HDATE HSTRY)
 A            HTIME     R        O  2 18REFFLD(HSRCD/HTIME HSTRY)
 A            WADDR1    R        O  5  1REFFLD(WRRCD/WADDR1 WRHS)
 A            DADDR1    R        O  5 42REFFLD(DSRCD/DADDR1   DSTRCT)
 A            WADDR2    R        O  6  1REFFLD(WRRCD/WADDR2 WRHS)
 A            DADDR2    R        O  6 42REFFLD(DSRCD/DADDR2   DSTRCT)
 A            WCITY     R        O  7  1REFFLD(WRRCD/WCITY   WRHS)
 A            WSTATE    R        O  7 22REFFLD(WRRCD/WSTATE WRHS)
 A            WZIP      R        O  7 25REFFLD(WRRCD/WZIP   WRHS)
 A            DCITY     R        O  7 42REFFLD(DSRCD/DCITY   DSTRCT)
 A            DSTATE    R        O  7 62REFFLD(DSRCD/DSTATE DSTRCT)
 A            DZIP      R        O  7 66REFFLD(DSRCD/DZIP   DSTRCT)
 A            CID       R        O  9 11REFFLD(CSRCD/CID    CSTMR)
 A            CFIRST    R        O 10  9REFFLD(CSRCD/CFIRST CSTMR)
 A            CINIT     R        O 10 26REFFLD(CSRCD/CINIT CSTMR)
 A            CLAST     R        O 10 29REFFLD(CSRCD/CLAST CSTMR)
 A            CLDATE    R        O 10 58REFFLD(CSRCD/CLDATE CSTMR)
 A            CADDR1    R        O 11  9REFFLD(CSRCD/CADDR1 CSTMR)
 A            CCREDT    R        O 11 58REFFLD(CSRCD/CCREDT CSTMR)
 A            CADDR2    R        O 12  9REFFLD(CSRCD/CADDR2 CSTMR)
 A            CDCT           4  2O 12 58EDTCDE(1)
 A            CCITY     R        O 13  9REFFLD(CSRCD/CCITY CSTMR)
 A            CSTATE    R        O 13 30REFFLD(CSRCD/CSTATE CSTMR)
 A            CZIP      R        O 13 33REFFLD(CSRCD/CZIP CSTMR)
 A            CPHONE         19A  O 13 58
 A            CBAL      R        O 15 55REFFLD(CSRCD/CBAL CSTMR)
 A            PAYMNT         6Y 2O 15 15EDTCDE(J $)
 A            CCRDLM    R        O 16 15REFFLD(CSRCD/CCRDLM CSTMR)
**************************************************************
***      ORDER STATUS DISPLAYS DSPREC30(INPUT) DSPREC35-45(OUTPUT)
**************************************************************
*A*  Order Status INPUT Screen
 A          R DSPREC30
 A                                      CLRL(22)
 A            HIDEME         2   H
 A                                    1 35'Order-Status'
 A                                      DSPATR(HI)
 A                                    2  1'Warehouse:'
 A            OWID      R        O  2 12REFFLD(ORRCD/OWID   ORDERS)
 A                                    2 19'District:'
 A            ODID      R        I  2 29REFFLD(DSPDID)
 A                                      CHECK(ME RZ)
 A                                    3  1'Customer:'
 A            OCID           4D  I  3 11
 A                                      CHECK(RZ)
 A                                    3 18'Name:'
 A            CLAST     R        I  3 44REFFLD(CSRCD/CLAST CSTMR)
 A                                    4  1'Cust-Balance:'
 A                                    6  1'Order Number:'
 A                                    6 26'Entry Date:'
 A                                    6 60'Carrier Number:'
 A                                    7  1'Supply W'
 A                                    7 14'Item Id'
 A                                    7 25'Qty'
 A                                    7 33'Amount'
 A                                    7 45'Delivery Date'
 A          R DSPREC31
 A                                      CLRL(*NO)
 A            HIDEME         2   H
 A                                   22 20'ENTER CID OR CLAST'
 A                                      DSPATR(HI)
 A*  ORDER STATUS OUTPUT SCREEN
 A          R DSPREC35                  CLRL(*NO)
 A            HIDEME         2   H
 A            OCID      R        O  3 11REFFLD(ORRCD/OCID   ORDERS)
 A            CFIRST    R        O  3 24REFFLD(CSRCD/CFIRST CSTMR)
 A            CINIT     R        O  3 41REFFLD(CSRCD/CINIT CSTMR)
 A            CLAST     R        O  3 44REFFLD(CSRCD/CLAST CSTMR)
 A            CBAL      R        O  4 15REFFLD(CSRCD/CBAL CSTMR)
 A            OID       R        O  6 15REFFLD(DSPOID)
 A            OENTDT    R        O  6 38REFFLD(ORRCD/OENTDT ORDERS)
 A            OENTTM    R        O  6 49REFFLD(ORRCD/OENTTM ORDERS)
 A            OCARID    R        O  6 76REFFLD(ORRCD/OCARID ORDERS)
 A            OLSPWH1   R        O  8  3REFFLD(OLRCD/OLSPWH ORDLIN)
 A            OLIID1    R        O  8 15REFFLD(OLRCD/OLIID ORDLIN)
 A            OLQTY1    R        O  8 25REFFLD(DSPQTY)
 A            OLAMNT1   R        O  8 31REFFLD(OLRCD/OLAMNT ORDLIN)
 A            OLDLVD1   R        O  8 47REFFLD(OLRCD/OLDLVD ORDLIN)
 A            OLSPWH2   R        O  9  3REFFLD(OLRCD/OLSPWH ORDLIN)
 A            OLIID2    R        O  9 15REFFLD(OLRCD/OLIID ORDLIN)
 A            OLQTY2    R        O  9 25REFFLD(DSPQTY)
 A            OLAMNT2   R        O  9 31REFFLD(OLRCD/OLAMNT ORDLIN)
 A            OLDLVD2   R        O  9 47REFFLD(OLRCD/OLDLVD ORDLIN)
 A            OLSPWH3   R        O 10  3REFFLD(OLRCD/OLSPWH ORDLIN)
 A            OLIID3    R        O 10 15REFFLD(OLRCD/OLIID ORDLIN)
 A            OLQTY3    R        O 10 25REFFLD(DSPQTY)
 A            OLAMNT3   R        O 10 31REFFLD(OLRCD/OLAMNT ORDLIN)
 A            OLDLVD3   R        O 10 47REFFLD(OLRCD/OLDLVD ORDLIN)
 A            OLSPWH4   R        O 11  3REFFLD(OLRCD/OLSPWH ORDLIN)
 A            OLIID4    R        O 11 15REFFLD(OLRCD/OLIID ORDLIN)
 A            OLQTY4    R        O 11 25REFFLD(DSPQTY)
 A            OLAMNT4   R        O 11 31REFFLD(OLRCD/OLAMNT ORDLIN)
 A            OLDLVD4   R        O 11 47REFFLD(OLRCD/OLDLVD ORDLIN)
 A            OLSPWH5   R        O 12  3REFFLD(OLRCD/OLSPWH ORDLIN)
 A            OLIID5    R        O 12 15REFFLD(OLRCD/OLIID ORDLIN)
 A            OLQTY5    R        O 12 25REFFLD(DSPQTY)
 A            OLAMNT5   R        O 12 31REFFLD(OLRCD/OLAMNT ORDLIN)
 A            OLDLVD5   R        O 12 47REFFLD(OLRCD/OLDLVD ORDLIN)
 A*  ORDER STATUS OUTPUT SCREEN
 A          R DSPREC36                  CLRL(*NO)
 A            HIDEME         2   H
 A            OCID      R        O  3 11REFFLD(ORRCD/OCID   ORDERS)
 A            CFIRST    R        O  3 24REFFLD(CSRCD/CFIRST CSTMR)
 A            CINIT     R        O  3 41REFFLD(CSRCD/CINIT CSTMR)
 A            CLAST     R        O  3 44REFFLD(CSRCD/CLAST CSTMR)
 A            CBAL      R        O  4 15REFFLD(CSRCD/CBAL  CSTMR)
 A            OID       R        O  6 15REFFLD(DSPOID)
 A            OENTDT    R        O  6 38REFFLD(ORRCD/OENTDT ORDERS)
 A            OENTTM    R        O  6 49REFFLD(ORRCD/OENTTM ORDERS)
 A            OCARID    R        O  6 76REFFLD(ORRCD/OCARID ORDERS)
 A            OLSPWH1   R        O  8  3REFFLD(OLRCD/OLSPWH ORDLIN)
 A            OLIID1    R        O  8 15REFFLD(OLRCD/OLIID ORDLIN)
 A            OLQTY1    R        O  8 25REFFLD(DSPQTY)
 A            OLAMNT1   R        O  8 31REFFLD(OLRCD/OLAMNT ORDLIN)
 A            OLDLVD1   R        O  8 47REFFLD(OLRCD/OLDLVD ORDLIN)
 A            OLSPWH2   R        O  9  3REFFLD(OLRCD/OLSPWH ORDLIN)
 A            OLIID2    R        O  9 15REFFLD(OLRCD/OLIID ORDLIN)
 A            OLQTY2    R        O  9 25REFFLD(DSPQTY)
 A            OLAMNT2   R        O  9 31REFFLD(OLRCD/OLAMNT ORDLIN)
 A            OLDLVD2   R        O  9 47REFFLD(OLRCD/OLDLVD ORDLIN)
 A            OLSPWH3   R        O 10  3REFFLD(OLRCD/OLSPWH ORDLIN)
 A            OLIID3    R        O 10 15REFFLD(OLRCD/OLIID ORDLIN)
 A            OLQTY3    R        O 10 25REFFLD(DSPQTY)
 A            OLAMNT3   R        O 10 31REFFLD(OLRCD/OLAMNT ORDLIN)
 A            OLDLVD3   R        O 10 47REFFLD(OLRCD/OLDLVD ORDLIN)
 A            OLSPWH4   R        O 11  3REFFLD(OLRCD/OLSPWH ORDLIN)
 A            OLIID4    R        O 11 15REFFLD(OLRCD/OLIID ORDLIN)
 A            OLQTY4    R        O 11 25REFFLD(DSPQTY)
 A            OLAMNT4   R        O 11 31REFFLD(OLRCD/OLAMNT ORDLIN)
 A            OLDLVD4   R        O 11 47REFFLD(OLRCD/OLDLVD ORDLIN)
 A            OLSPWH5   R        O 12  3REFFLD(OLRCD/OLSPWH ORDLIN)
 A            OLIID5    R        O 12 15REFFLD(OLRCD/OLIID ORDLIN)
 A            OLQTY5    R        O 12 25REFFLD(DSPQTY)
```

```
A            OLAMNT5   R        O 12 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD5   R        O 12 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH6   R        O 13  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID6    R        O 13 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY6    R        O 13 25REFFLD(DSPQTY)
A            OLAMNT6   R        O 13 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD6   R        O 13 47REFFLD(OLRCD/OLDLVD ORDLIN)
A*  ORDER STATUS OUTPUT SCREEN
A          R DSPREC37                    CLRL(*NO)
A            HIDEME          2   H
A            OCID      R         O  3 11REFFLD(ORRCD/OCID   ORDERS)
A            CFIRST    R         O  3 24REFFLD(CSRCD/CFIRST CSTMR)
A            CINIT     R         O  3 41REFFLD(CSRCD/CINIT CSTMR)
A            CLAST     R         O  3 44REFFLD(CSRCD/CLAST CSTMR)
A            CBAL      R         O  4 15REFFLD(CSRCD/CBAL  CSTMR)
A            OID       R         O  6 15REFFLD(DSPOID)
A            OENTDT    R         O  6 38REFFLD(ORRCD/OENTDT ORDERS)
A            OENTTM    R         O  6 49REFFLD(ORRCD/OENTTM ORDERS)
A            OCARID    R         O  6 76REFFLD(ORRCD/OCARID ORDERS)
A            OLSPWH1   R         O  8  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID1    R         O  8 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY1    R         O  8 25REFFLD(DSPQTY)
A            OLAMNT1   R         O  8 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD1   R         O  8 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH2   R         O  9  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID2    R         O  9 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY2    R         O  9 25REFFLD(DSPQTY)
A            OLAMNT2   R         O  9 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD2   R         O  9 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH3   R         O 10  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID3    R         O 10 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY3    R         O 10 25REFFLD(DSPQTY)
A            OLAMNT3   R         O 10 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD3   R         O 10 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH4   R         O 11  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID4    R         O 11 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY4    R         O 11 25REFFLD(DSPQTY)
A            OLAMNT4   R         O 11 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD4   R         O 11 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH5   R         O 12  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID5    R         O 12 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY5    R         O 12 25REFFLD(DSPQTY)
A            OLAMNT5   R         O 12 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD5   R         O 12 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH6   R         O 13  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID6    R         O 13 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY6    R         O 13 25REFFLD(DSPQTY)
A            OLAMNT6   R         O 13 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD6   R         O 13 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH7   R         O 14  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID7    R         O 14 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY7    R         O 14 25REFFLD(DSPQTY)
A            OLAMNT7   R         O 14 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD7   R         O 14 47REFFLD(OLRCD/OLDLVD ORDLIN)
A*  ORDER STATUS OUTPUT SCREEN
A          R DSPREC38                    CLRL(*NO)
A            HIDEME          2   H
A            OCID      R         O  3 11REFFLD(ORRCD/OCID   ORDERS)
A            CFIRST    R         O  3 24REFFLD(CSRCD/CFIRST CSTMR)
A            CINIT     R         O  3 41REFFLD(CSRCD/CINIT CSTMR)
A            CLAST     R         O  3 44REFFLD(CSRCD/CLAST CSTMR)
A            CBAL      R         O  4 15REFFLD(CSRCD/CBAL  CSTMR)
A            OID       R         O  6 15REFFLD(DSPOID)
A            OENTDT    R         O  6 38REFFLD(ORRCD/OENTDT ORDERS)
A            OENTTM    R         O  6 49REFFLD(ORRCD/OENTTM ORDERS)
A            OCARID    R         O  6 76REFFLD(ORRCD/OCARID ORDERS)
A            OLSPWH1   R         O  8  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID1    R         O  8 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY1    R         O  8 25REFFLD(DSPQTY)
A            OLAMNT1   R         O  8 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD1   R         O  8 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH2   R         O  9  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID2    R         O  9 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY2    R         O  9 25REFFLD(DSPQTY)
A            OLAMNT2   R         O  9 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD2   R         O  9 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH3   R         O 10  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID3    R         O 10 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY3    R         O 10 25REFFLD(DSPQTY)
A            OLAMNT3   R         O 10 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD3   R         O 10 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH4   R         O 11  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID4    R         O 11 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY4    R         O 11 25REFFLD(DSPQTY)
A            OLAMNT4   R         O 11 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD4   R         O 11 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH5   R         O 12  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID5    R         O 12 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY5    R         O 12 25REFFLD(DSPQTY)
A            OLAMNT5   R         O 12 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD5   R         O 12 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH6   R         O 13  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID6    R         O 13 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY6    R         O 13 25REFFLD(DSPQTY)
A            OLAMNT6   R         O 13 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD6   R         O 13 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH7   R         O 14  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID7    R         O 14 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY7    R         O 14 25REFFLD(DSPQTY)
A            OLAMNT7   R         O 14 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD7   R         O 14 47REFFLD(OLRCD/OLDLVD ORDLIN)
A*  ORDER STATUS OUTPUT SCREEN
A          R DSPREC39                    CLRL(*NO)
A            HIDEME          2   H
A            OCID      R         O  3 11REFFLD(ORRCD/OCID   ORDERS)
A            CFIRST    R         O  3 24REFFLD(CSRCD/CFIRST CSTMR)
A            CINIT     R         O  3 41REFFLD(CSRCD/CINIT CSTMR)
A            CLAST     R         O  3 44REFFLD(CSRCD/CLAST CSTMR)
A            CBAL      R         O  4 15REFFLD(CSRCD/CBAL CSTMR)
A            OID       R         O  6 15REFFLD(DSPOID)
A            OENTDT    R         O  6 38REFFLD(ORRCD/OENTDT ORDERS)
A            OENTTM    R         O  6 49REFFLD(ORRCD/OENTTM ORDERS)
A            OCARID    R         O  6 76REFFLD(ORRCD/OCARID ORDERS)
A            OLSPWH1   R         O  8  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID1    R         O  8 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY1    R         O  8 25REFFLD(DSPQTY)
A            OLAMNT1   R         O  8 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD1   R         O  8 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH2   R         O  9  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID2    R         O  9 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY2    R         O  9 25REFFLD(DSPQTY)
A            OLAMNT2   R         O  9 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD2   R         O  9 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH3   R         O 10  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID3    R         O 10 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY3    R         O 10 25REFFLD(DSPQTY)
A            OLAMNT3   R         O 10 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD3   R         O 10 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH4   R         O 11  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID4    R         O 11 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY4    R         O 11 25REFFLD(DSPQTY)
A            OLAMNT4   R         O 11 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD4   R         O 11 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH5   R         O 12  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID5    R         O 12 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY5    R         O 12 25REFFLD(DSPQTY)
A            OLAMNT5   R         O 12 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD5   R         O 12 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH6   R         O 13  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID6    R         O 13 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY6    R         O 13 25REFFLD(DSPQTY)
A            OLAMNT6   R         O 13 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD6   R         O 13 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH7   R         O 14  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID7    R         O 14 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY7    R         O 14 25REFFLD(DSPQTY)
A            OLAMNT7   R         O 14 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD7   R         O 14 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH8   R         O 15  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID8    R         O 15 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY8    R         O 15 25REFFLD(DSPQTY)
A            OLAMNT8   R         O 15 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD8   R         O 15 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH9   R         O 16  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID9    R         O 16 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY9    R         O 16 25REFFLD(DSPQTY)
A            OLAMNT9   R         O 16 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD9   R         O 16 47REFFLD(OLRCD/OLDLVD ORDLIN)
A*  ORDER STATUS OUTPUT SCREEN
A          R DSPREC40                    CLRL(*NO)
A            HIDEME          2   H
A            OCID      R         O  3 11REFFLD(ORRCD/OCID   ORDERS)
A            CFIRST    R         O  3 24REFFLD(CSRCD/CFIRST CSTMR)
A            CINIT     R         O  3 41REFFLD(CSRCD/CINIT CSTMR)
A            CLAST     R         O  3 44REFFLD(CSRCD/CLAST CSTMR)
A            CBAL      R         O  4 15REFFLD(CSRCD/CBAL  CSTMR)
A            OID       R         O  6 15REFFLD(DSPOID)
A            OENTDT    R         O  6 38REFFLD(ORRCD/OENTDT ORDERS)
A            OENTTM    R         O  6 49REFFLD(ORRCD/OENTTM ORDERS)
A            OCARID    R         O  6 76REFFLD(ORRCD/OCARID ORDERS)
A            OLSPWH1   R         O  8  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID1    R         O  8 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY1    R         O  8 25REFFLD(DSPQTY)
A            OLAMNT1   R         O  8 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD1   R         O  8 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH2   R         O  9  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID2    R         O  9 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY2    R         O  9 25REFFLD(DSPQTY)
A            OLAMNT2   R         O  9 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD2   R         O  9 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH3   R         O 10  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID3    R         O 10 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY3    R         O 10 25REFFLD(DSPQTY)
A            OLAMNT3   R         O 10 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD3   R         O 10 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH4   R         O 11  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID4    R         O 11 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY4    R         O 11 25REFFLD(DSPQTY)
A            OLAMNT4   R         O 11 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD4   R         O 11 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH5   R         O 12  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID5    R         O 12 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY5    R         O 12 25REFFLD(DSPQTY)
A            OLAMNT5   R         O 12 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD5   R         O 12 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH6   R         O 13  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID6    R         O 13 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY6    R         O 13 25REFFLD(DSPQTY)
A            OLAMNT6   R         O 13 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD6   R         O 13 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH7   R         O 14  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID7    R         O 14 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY7    R         O 14 25REFFLD(DSPQTY)
A            OLAMNT7   R         O 14 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD7   R         O 14 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH8   R         O 15  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID8    R         O 15 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY8    R         O 15 25REFFLD(DSPQTY)
A            OLAMNT8   R         O 15 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD8   R         O 15 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH9   R         O 16  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID9    R         O 16 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY9    R         O 16 25REFFLD(DSPQTY)
A            OLAMNT9   R         O 16 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD9   R         O 16 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH10  R         O 17  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID10   R         O 17 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY10   R         O 17 25REFFLD(DSPQTY)
A            OLAMNT10  R         O 17 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD10  R         O 17 47REFFLD(OLRCD/OLDLVD ORDLIN)
A*  ORDER STATUS OUTPUT SCREEN
A          R DSPREC41                    CLRL(*NO)
A            HIDEME          2   H
A            OCID      R         O  3 11REFFLD(ORRCD/OCID   ORDERS)
A            CFIRST    R         O  3 24REFFLD(CSRCD/CFIRST CSTMR)
A            CINIT     R         O  3 41REFFLD(CSRCD/CINIT CSTMR)
A            CLAST     R         O  3 44REFFLD(CSRCD/CLAST CSTMR)
A            CBAL      R         O  4 15REFFLD(CSRCD/CBAL  CSTMR)
A            OID       R         O  6 15REFFLD(DSPOID)
A            OENTDT    R         O  6 38REFFLD(ORRCD/OENTDT ORDERS)
A            OENTTM    R         O  6 49REFFLD(ORRCD/OENTTM ORDERS)
A            OCARID    R         O  6 76REFFLD(ORRCD/OCARID ORDERS)
A            OLSPWH1   R         O  8  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID1    R         O  8 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY1    R         O  8 25REFFLD(DSPQTY)
A            OLAMNT1   R         O  8 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD1   R         O  8 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH2   R         O  9  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID2    R         O  9 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY2    R         O  9 25REFFLD(DSPQTY)
A            OLAMNT2   R         O  9 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD2   R         O  9 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH3   R         O 10  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID3    R         O 10 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY3    R         O 10 25REFFLD(DSPQTY)
A            OLAMNT3   R         O 10 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD3   R         O 10 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH4   R         O 11  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID4    R         O 11 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY4    R         O 11 25REFFLD(DSPQTY)
A            OLAMNT4   R         O 11 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD4   R         O 11 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH5   R         O 12  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID5    R         O 12 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY5    R         O 12 25REFFLD(DSPQTY)
A            OLAMNT5   R         O 12 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD5   R         O 12 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH6   R         O 13  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID6    R         O 13 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY6    R         O 13 25REFFLD(DSPQTY)
A            OLAMNT6   R         O 13 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD6   R         O 13 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH7   R         O 14  3REFFLD(OLRCD/OLSPWH ORDLIN)
```

```
A            OLIID7    R        O 14 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY7    R        O 14 25REFFLD(DSPQTY)
A            OLAMNT7   R        O 14 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD7   R        O 14 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH8   R        O 15  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID8    R        O 15 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY8    R        O 15 25REFFLD(DSPQTY)
A            OLAMNT8   R        O 15 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD8   R        O 15 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH9   R        O 16  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID9    R        O 16 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY9    R        O 16 25REFFLD(DSPQTY)
A            OLAMNT9   R        O 16 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD9   R        O 16 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH10  R        O 17  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID10   R        O 17 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY10   R        O 17 25REFFLD(DSPQTY)
A            OLAMNT10  R        O 17 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD10  R        O 17 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH11  R        O 18  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID11   R        O 18 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY11   R        O 18 25REFFLD(DSPQTY)
A            OLAMNT11  R        O 18 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD11  R        O 18 47REFFLD(OLRCD/OLDLVD ORDLIN)
A*  ORDER STATUS OUTPUT SCREEN
A          R DSPREC42                CLRL(*NO)
A            HIDEME         2   H
A            OCID      R        O  3 11REFFLD(ORRCD/OCID   ORDERS)
A            CFIRST    R        O  3 24REFFLD(CSRCD/CFIRST CSTMR)
A            CINIT     R        O  3 41REFFLD(CSRCD/CINIT CSTMR)
A            CLAST     R        O  3 44REFFLD(CSRCD/CLAST CSTMR)
A            CBAL      R        O  4 15REFFLD(CSRCD/CBAL  CSTMR)
A            OID       R        O  6 15REFFLD(DSPOID)
A            OENTDT    R        O  6 38REFFLD(ORRCD/OENTDT ORDERS)
A            OENTTM    R        O  6 49REFFLD(ORRCD/OENTTM ORDERS)
A            OCARID    R        O  6 76REFFLD(ORRCD/OCARID ORDERS)
A            OLSPWH1   R        O  8  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID1    R        O  8 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY1    R        O  8 25REFFLD(DSPQTY)
A            OLAMNT1   R        O  8 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD1   R        O  8 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH2   R        O  9  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID2    R        O  9 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY2    R        O  9 25REFFLD(DSPQTY)
A            OLAMNT2   R        O  9 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD2   R        O  9 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH3   R        O 10  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID3    R        O 10 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY3    R        O 10 25REFFLD(DSPQTY)
A            OLAMNT3   R        O 10 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD3   R        O 10 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH4   R        O 11  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID4    R        O 11 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY4    R        O 11 25REFFLD(DSPQTY)
A            OLAMNT4   R        O 11 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD4   R        O 11 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH5   R        O 12  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID5    R        O 12 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY5    R        O 12 25REFFLD(DSPQTY)
A            OLAMNT5   R        O 12 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD5   R        O 12 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH6   R        O 13  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID6    R        O 13 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY6    R        O 13 25REFFLD(DSPQTY)
A            OLAMNT6   R        O 13 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD6   R        O 13 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH7   R        O 14  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID7    R        O 14 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY7    R        O 14 25REFFLD(DSPQTY)
A            OLAMNT7   R        O 14 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD7   R        O 14 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH8   R        O 15  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID8    R        O 15 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY8    R        O 15 25REFFLD(DSPQTY)
A            OLAMNT8   R        O 15 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD8   R        O 15 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH9   R        O 16  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID9    R        O 16 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY9    R        O 16 25REFFLD(DSPQTY)
A            OLAMNT9   R        O 16 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD9   R        O 16 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH10  R        O 17  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID10   R        O 17 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY10   R        O 17 25REFFLD(DSPQTY)
A            OLAMNT10  R        O 17 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD10  R        O 17 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH11  R        O 18  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID11   R        O 18 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY11   R        O 18 25REFFLD(DSPQTY)
A            OLAMNT11  R        O 18 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD11  R        O 18 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH12  R        O 19  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID12   R        O 19 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY12   R        O 19 25REFFLD(DSPQTY)
A            OLAMNT12  R        O 19 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD12  R        O 19 47REFFLD(OLRCD/OLDLVD ORDLIN)
A*  ORDER STATUS OUTPUT SCREEN
A          R DSPREC43                CLRL(*NO)
A            HIDEME         2   H
A            OCID      R        O  3 11REFFLD(ORRCD/OCID   ORDERS)
A            CFIRST    R        O  3 24REFFLD(CSRCD/CFIRST CSTMR)
A            CINIT     R        O  3 41REFFLD(CSRCD/CINIT CSTMR)
A            CLAST     R        O  3 44REFFLD(CSRCD/CLAST CSTMR)
A            CBAL      R        O  4 15REFFLD(CSRCD/CBAL  CSTMR)
A            OID       R        O  6 15REFFLD(DSPOID)
A            OENTDT    R        O  6 38REFFLD(ORRCD/OENTDT ORDERS)
A            OENTTM    R        O  6 49REFFLD(ORRCD/OENTTM ORDERS)
A            OCARID    R        O  6 76REFFLD(ORRCD/OCARID ORDERS)
A            OLSPWH1   R        O  8  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID1    R        O  8 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY1    R        O  8 25REFFLD(DSPQTY)
A            OLAMNT1   R        O  8 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD1   R        O  8 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH2   R        O  9  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID2    R        O  9 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY2    R        O  9 25REFFLD(DSPQTY)
A            OLAMNT2   R        O  9 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD2   R        O  9 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH3   R        O 10  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID3    R        O 10 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY3    R        O 10 25REFFLD(DSPQTY)
A            OLAMNT3   R        O 10 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD3   R        O 10 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH4   R        O 11  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID4    R        O 11 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY4    R        O 11 25REFFLD(DSPQTY)
A            OLAMNT4   R        O 11 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD4   R        O 11 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH5   R        O 12  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID5    R        O 12 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY5    R        O 12 25REFFLD(DSPQTY)
A            OLAMNT5   R        O 12 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD5   R        O 12 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH6   R        O 13  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID6    R        O 13 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY6    R        O 13 25REFFLD(DSPQTY)
A            OLAMNT6   R        O 13 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD6   R        O 13 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH7   R        O 14  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID7    R        O 14 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY7    R        O 14 25REFFLD(DSPQTY)
A            OLAMNT7   R        O 14 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD7   R        O 14 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH8   R        O 15  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID8    R        O 15 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY8    R        O 15 25REFFLD(DSPQTY)
A            OLAMNT8   R        O 15 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD8   R        O 15 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH9   R        O 16  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID9    R        O 16 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY9    R        O 16 25REFFLD(DSPQTY)
A            OLAMNT9   R        O 16 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD9   R        O 16 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH10  R        O 17  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID10   R        O 17 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY10   R        O 17 25REFFLD(DSPQTY)
A            OLAMNT10  R        O 17 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD10  R        O 17 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH11  R        O 18  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID11   R        O 18 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY11   R        O 18 25REFFLD(DSPQTY)
A            OLAMNT11  R        O 18 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD11  R        O 18 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH12  R        O 19  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID12   R        O 19 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY12   R        O 19 25REFFLD(DSPQTY)
A            OLAMNT12  R        O 19 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD12  R        O 19 47REFFLD(OLRCD/OLDLVD ORDLIN)
A*  ORDER STATUS OUTPUT SCREEN
A          R DSPREC44                CLRL(*NO)
A            HIDEME         2   H
A            OCID      R        O  3 11REFFLD(ORRCD/OCID   ORDERS)
A            CFIRST    R        O  3 24REFFLD(CSRCD/CFIRST CSTMR)
A            CINIT     R        O  3 41REFFLD(CSRCD/CINIT CSTMR)
A            CLAST     R        O  3 44REFFLD(CSRCD/CLAST CSTMR)
A            CBAL      R        O  4 15REFFLD(CSRCD/CBAL  CSTMR)
A            OID       R        O  6 15REFFLD(DSPOID)
A            OENTDT    R        O  6 38REFFLD(ORRCD/OENTDT ORDERS)
A            OENTTM    R        O  6 49REFFLD(ORRCD/OENTTM ORDERS)
A            OCARID    R        O  6 76REFFLD(ORRCD/OCARID ORDERS)
A            OLSPWH1   R        O  8  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID1    R        O  8 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY1    R        O  8 25REFFLD(DSPQTY)
A            OLAMNT1   R        O  8 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD1   R        O  8 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH2   R        O  9  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID2    R        O  9 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY2    R        O  9 25REFFLD(DSPQTY)
A            OLAMNT2   R        O  9 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD2   R        O  9 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH3   R        O 10  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID3    R        O 10 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY3    R        O 10 25REFFLD(DSPQTY)
A            OLAMNT3   R        O 10 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD3   R        O 10 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH4   R        O 11  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID4    R        O 11 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY4    R        O 11 25REFFLD(DSPQTY)
A            OLAMNT4   R        O 11 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD4   R        O 11 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH5   R        O 12  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID5    R        O 12 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY5    R        O 12 25REFFLD(DSPQTY)
A            OLAMNT5   R        O 12 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD5   R        O 12 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH6   R        O 13  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID6    R        O 13 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY6    R        O 13 25REFFLD(DSPQTY)
A            OLAMNT6   R        O 13 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD6   R        O 13 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH7   R        O 14  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID7    R        O 14 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY7    R        O 14 25REFFLD(DSPQTY)
A            OLAMNT7   R        O 14 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD7   R        O 14 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH8   R        O 15  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID8    R        O 15 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY8    R        O 15 25REFFLD(DSPQTY)
A            OLAMNT8   R        O 15 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD8   R        O 15 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH9   R        O 16  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID9    R        O 16 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY9    R        O 16 25REFFLD(DSPQTY)
A            OLAMNT9   R        O 16 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD9   R        O 16 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH10  R        O 17  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID10   R        O 17 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY10   R        O 17 25REFFLD(DSPQTY)
A            OLAMNT10  R        O 17 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD10  R        O 17 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH11  R        O 18  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID11   R        O 18 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY11   R        O 18 25REFFLD(DSPQTY)
A            OLAMNT11  R        O 18 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD11  R        O 18 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH12  R        O 19  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID12   R        O 19 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY12   R        O 19 25REFFLD(DSPQTY)
A            OLAMNT12  R        O 19 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD12  R        O 19 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH13  R        O 20  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID13   R        O 20 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY13   R        O 20 25REFFLD(DSPQTY)
A            OLAMNT13  R        O 20 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD13  R        O 20 47REFFLD(OLRCD/OLDLVD ORDLIN)
A*  ORDER STATUS OUTPUT SCREEN
A          R DSPREC45                CLRL(*NO)
A            HIDEME         2   H
A            OCID      R        O  3 11REFFLD(ORRCD/OCID   ORDERS)
A            CFIRST    R        O  3 24REFFLD(CSRCD/CFIRST CSTMR)
A            CINIT     R        O  3 41REFFLD(CSRCD/CINIT CSTMR)
A            CLAST     R        O  3 44REFFLD(CSRCD/CLAST CSTMR)
A            CBAL      R        O  4 15REFFLD(CSRCD/CBAL  CSTMR)
A            OID       R        O  6 15REFFLD(DSPOID)
A            OENTDT    R        O  6 38REFFLD(ORRCD/OENTDT ORDERS)
A            OENTTM    R        O  6 49REFFLD(ORRCD/OENTTM ORDERS)
A            OCARID    R        O  6 76REFFLD(ORRCD/OCARID ORDERS)
```

```
A            OLSPWH1   R        O  8  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID1    R        O  8 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY1    R        O  8 25REFFLD(DSPQTY)
A            OLAMNT1   R        O  8 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD1   R        O  8 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH2   R        O  9  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID2    R        O  9 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY2    R        O  9 25REFFLD(DSPQTY)
A            OLAMNT2   R        O  9 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD2   R        O  9 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH3   R        O 10  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID3    R        O 10 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY3    R        O 10 25REFFLD(DSPQTY)
A            OLAMNT3   R        O 10 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD3   R        O 10 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH4   R        O 11  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID4    R        O 11 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY4    R        O 11 25REFFLD(DSPQTY)
A            OLAMNT4   R        O 11 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD4   R        O 11 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH5   R        O 12  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID5    R        O 12 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY5    R        O 12 25REFFLD(DSPQTY)
A            OLAMNT5   R        O 12 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD5   R        O 12 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH6   R        O 13  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID6    R        O 13 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY6    R        O 13 25REFFLD(DSPQTY)
A            OLAMNT6   R        O 13 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD6   R        O 13 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH7   R        O 14  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID7    R        O 14 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY7    R        O 14 25REFFLD(DSPQTY)
A            OLAMNT7   R        O 14 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD7   R        O 14 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH8   R        O 15  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID8    R        O 15 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY8    R        O 15 25REFFLD(DSPQTY)
A            OLAMNT8   R        O 15 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD8   R        O 15 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH9   R        O 16  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID9    R        O 16 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY9    R        O 16 25REFFLD(DSPQTY)
A            OLAMNT9   R        O 16 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD9   R        O 16 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH10  R        O 17  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID10   R        O 17 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY10   R        O 17 25REFFLD(DSPQTY)
A            OLAMNT10  R        O 17 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD10  R        O 17 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH11  R        O 18  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID11   R        O 18 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY11   R        O 18 25REFFLD(DSPQTY)
A            OLAMNT11  R        O 18 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD11  R        O 18 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH12  R        O 19  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID12   R        O 19 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY12   R        O 19 25REFFLD(DSPQTY)
A            OLAMNT12  R        O 19 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD12  R        O 19 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH13  R        O 20  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID13   R        O 20 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY13   R        O 20 25REFFLD(DSPQTY)
A            OLAMNT13  R        O 20 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD13  R        O 20 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH14  R        O 21  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID14   R        O 21 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY14   R        O 21 25REFFLD(DSPQTY)
A            OLAMNT14  R        O 21 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD14  R        O 21 47REFFLD(OLRCD/OLDLVD ORDLIN)
A            OLSPWH15  R        O 22  3REFFLD(OLRCD/OLSPWH ORDLIN)
A            OLIID15   R        O 22 15REFFLD(OLRCD/OLIID ORDLIN)
A            OLQTY15   R        O 22 25REFFLD(DSPQTY)
A            OLAMNT15  R        O 22 31REFFLD(OLRCD/OLAMNT ORDLIN)
A            OLDLVD15  R        O 22 47REFFLD(OLRCD/OLDLVD ORDLIN)
A**************************************************************
A* DELIVERY INPUT SCREEN
A**************************************************************
A          R DSPREC22
A                                     CLRL(22)
A                                  1 38'Delivery'
A                                     DSPATR(HI)
A                                  2  1'Warehouse:'
A            WID       R        O  2 12REFFLD(ORRCD/OWID   ORDERS)
A                                  4  1'Carrier Number:'
A                                     DSPATR(HI)
A            OCARID         2D  I  4 17
A                                     CHECK(RZ ME)
A**************************************************************
A*  DELIVERY RESPONSE SCREEN
A**************************************************************
A          R DSPREC23                 CLRL(*NO)
A                                  6  1'Execution Status:'
A                                  6 19'Delivery has been queued'
*A*  Stock Level INPUT SCREEN
A          R DSPREC50
A*%%TS  SD  19921002  172130  PPCOC       REL-V2R2M0  5738-PW1
A                                     CLRL(22)
A                                  1 35'Stock-Level'
A                                  2  1'Warehouse:'
A            DWID      R        O  2 12REFFLD(DSRCD/DWID   DSTRCT)
A                                  2 19'District:'
A            DID       R        O  2 29REFFLD(DSPDID)
A                                  4  1'Stock Level Threshold:'
A            THRSH          2Y 0I  4 24CHECK(ME RZ)
A                                  6  1'low stock found:'
*A*  Stock Level Screen
A          R DSPREC51                 CLRL(*NO)
A            BLWSTK         3Y 0O  6 18EDTWRD('0  ')
A          R DSPRECID
A*%%TS  SD  19920819  111811  PPCOC       REL-V2R2M0  5738-PW1
A                                     CLRL(*NO)
A            HIDEME         2   H
A            CID       R        O  3 12REFFLD(CSRCD/CID   CSTMR)
A            CLAST     R        O  3 25REFFLD(CSRCD/CLAST CSTMR)
A            CCREDT    R        O  3 52REFFLD(CSRCD/CCREDT CSTMR)
A            OID       R        O  4 15REFFLD(DSPOID)
A                                 22 20'INPUT DATA  IS NOT VALID'
A                                     DSPATR(HI)
A          R DSPREC77
A                                     CLRL(*NO)
A            HIDEME         2   H
A                                 22 20'INPUT DATA  IS NOT VALID'
A                                     DSPATR(HI)
A          R DSPREC62
A                                     CLRL(*NO)
A            HIDEME         2   H
A                                 22 20'INPUT DATA  IS NOT VALID'
A                                     DSPATR(HI)


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

DLVRICFF: Delivery ICF File on Client

```
A**************************************************************
A*                                                            *
A*            ICF file for DELIVERY transaction               *
A*                                                            *
A**************************************************************
A*   File level indicators:
A*
A                                     INDARA
A*
A**************************************************************
A*            ICF RECORD FORMATS                              *
A**************************************************************
A          R RCVDATA                  TEXT('DATA RECEIVED FROM -
A                                     REMOTE SYSTEM.')
A            RCVFLD       670A       TEXT('NEW ORD RCV FIELD')
A          R SNDDATA                  TEXT('DATA SENT TO THE   -
A                                     REMOTE SYSTEM.')
A  10                                 FRCDTA
A            SNDFLD       230A       TEXT('TRANSACTION DATA')
A          R EVOKPGM                  TEXT('EVOKE THE TARGET   -
A                                     PROGRAM.')
A                                     SECURITY(3 *USER)
A                                     EVOKE(TPCCNCSRVR/ATPCCMTRD)


A          R ENDREC
```

MAINICFF: Neworder / Payment / Order Status ICF Files on Client

```
A**************************************************************
A*                                                            *
A*            ICF FILE FOR MAIN TRANSACTIONS                  *
A*                                                            *
A**************************************************************
A*   File level indicators:
A*
A                                     INDARA
A*
A                                     RCVTRNRND(15 'END OF DATA')
A*
A  30                                 DETACH
A*
A                                     INDTXT(30 '30->DETACH TARG-
A                                     ET PROGRAM.')
A*
A                                     RCVDETACH(35 'RECEIVED    -
A                                     DETACHED.')
A*
A*
A**************************************************************
A*            ICF RECORD FORMATS                              *
A**************************************************************
A          R RCVDATA                  TEXT('DATA RECEIVED FROM -
A                                     REMOTE SYSTEM.')
A            RCVFLD       670A       TEXT('NEW ORD RCV FIELD')
A          R SNDDATA                  TEXT('DATA SENT TO THE   -
A                                     REMOTE SYSTEM.')
A                                     INVITE
A            SNDFLD       230A       TEXT('TRANSACTION DATA')
A          R EVOKPGM                  TEXT('EVOKE THE TARGET   -
A                                     PROGRAM.')
A                                     EVOKE(TPCCNCSRVR/ATPCCMTRA)
A                                     SECURITY(3 *USER)


A          R ENDREC
```

STOCICFF: Stock Level ICF File on Client

```
A**************************************************************
A*                                                            *
A*            ICF file for DELIVERY transaction               *
A*                                                            *
A**************************************************************
A*   File level indicators:
A*
A                                     INDARA
A*
A                                     RCVTRNRND(15 'END OF DATA')
A*
A  30                                 DETACH
A*
A                                     INDTXT(30 '30->DETACH TARG-
A                                     ET PROGRAM.')
A*
A                                     RCVDETACH(35 'RECEIVED    -
A                                     DETACHED.')
A*
A*
A**************************************************************
A*            ICF RECORD FORMATS                              *
A**************************************************************
A          R RCVDATA                  TEXT('DATA RECEIVED FROM -
A                                     REMOTE SYSTEM.')
A            RCVFLD       670A       TEXT('NEW ORD RCV FIELD')
A          R SNDDATA                  TEXT('DATA SENT TO THE   -
A                                     REMOTE SYSTEM.')
A                                     INVITE
```

```
A            SNDFLD       230A         TEXT('TRANSACTION DATA')
A          R EVOKPGM                   TEXT('EVOKE THE TARGET   -
A                                      PROGRAM.')
A                                      SECURITY(3 *USER)
A                                      EVOKE(TPCCNCSRVR/ATPCCMTRS)


A          R ENDREC
```

ATPCCMTRD: Delivery Start Program

```
        PGM

        DCL        VAR(&DATALIB) TYPE(*CHAR) LEN(10) +
                     VALUE(TPCCD697)

        ADDLIBLE   LIB(&DATALIB)
        MONMSG CPF2103
        CHGJOB     RUNPTY(34)
        STRCMTCTL  LCKLVL(*ALL)

        OVRDBF     FILE(ORDERSLF) TOFILE(ORDERSLF) +
                     LVLCHK(*NO) SHARE(*NO)

        OVRDBF     FILE(NEWORDLF) TOFILE(NEWORDLF) +
                     LVLCHK(*NO) SHARE(*NO) NBRRCDS(10)

        OVRDBF     FILE(CSTMR) TOFILE(CSTMR) LVLCHK(*NO) +
                     SHARE(*NO)

        OVRDBF     FILE(DLVRYLOG) TOFILE(DLVRYLOG) LVLCHK(*NO) +
                     SHARE(*NO)

        OVRDBF     FILE(ORDLINLF) TOFILE(ORDLINLF) +
                     LVLCHK(*NO) SHARE(*NO) +
                     SEQONLY(*NO) NBRRCDS(120)

        CALL       PGM(TPCCNCSRVR/DLVRYMTR)

        ENDCMTCTL


        ENDPGM
```

DLVRYMTR: Batch Portion of Delivery Transaction

```
 PROCESS NOTRUNC NORANGE.
 IDENTIFICATION DIVISION.
   PROGRAM-ID. DLVRY.
   AUTHOR.  PPCOC.
   INSTALLATION. IBM-ROCHESTER.
   DATE-WRITTEN.
   DATE-COMPILED.
*****************************************************************
*   C  H  A  N  G  E  S     &      U  P  D  A  T  E  S
*
*   LATEST CHANGES LISTED FIRST
*
*   MM/DD/YY   AUTHOR NAME          LINES CHANGED/ADDED: NNN
*
*
*     DESCRIPTION OF CHANGE:
*
*
*
*****************************************************************
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
   SOURCE-COMPUTER. IBM-AS400.
*  SOURCE-COMPUTER. IBM-AS400 WITH DEBUGGING MODE.
   OBJECT-COMPUTER. IBM-AS400.
 INPUT-OUTPUT SECTION.
 FILE-CONTROL.
       SELECT DLVRICFF ASSIGN TO WORKSTATION-DLVRICFF-SI
              ORGANIZATION IS TRANSACTION
              FILE STATUS IS STATUS-IND MAJ-MIN.
     SELECT NEWORD
            ASSIGN TO DATABASE-NEWORDLF
            ORGANIZATION IS INDEXED
            ACCESS MODE IS DYNAMIC
            RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
            WITH DUPLICATES
            FILE STATUS IS NEWORD-FILE-STATUS.
     SELECT DISTRICT-FILE
           ASSIGN TO DATABASE-DSTRCT
           ORGANIZATION IS INDEXED
           ACCESS MODE IS RANDOM
           RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
           WITH DUPLICATES.
     SELECT ORDERS
            ASSIGN TO DATABASE-ORDERSLF
            ORGANIZATION IS INDEXED
            ACCESS MODE IS RANDOM
            RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
            WITH DUPLICATES.
     SELECT ORDERLINE
            ASSIGN TO DATABASE-ORDLINLF
            ORGANIZATION IS INDEXED
            ACCESS MODE IS DYNAMIC
            RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
            WITH DUPLICATES.
     SELECT CSTMR
            ASSIGN TO DATABASE-CSTMR
            ORGANIZATION IS INDEXED
            ACCESS MODE IS RANDOM
            RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
            WITH DUPLICATES.
     SELECT DLVRYLOG
            ASSIGN TO DATABASE-DLVRYLOG
            ORGANIZATION IS SEQUENTIAL.
D    SELECT ISO-IFILE
D           ASSIGN TO DATABASE-ISOIFILE
D           ORGANIZATION IS SEQUENTIAL
D           ACCESS IS SEQUENTIAL.
D    SELECT DBUG-FILE
D           ASSIGN TO FORMATFILE-DBUGPRT2
D           ORGANIZATION IS SEQUENTIAL
D           ACCESS IS SEQUENTIAL.
 I-O-CONTROL.
     COMMITMENT CONTROL FOR ORDERS ORDERLINE
                        DISTRICT-FILE CSTMR NEWORD.

 DATA DIVISION.
 FILE SECTION.
   FD  DLVRICFF
       LABEL RECORDS ARE STANDARD.
   01  ICFREC.
       COPY DDS-ALL-FORMATS OF DLVRICFF.

   05 DLVRY-O REDEFINES DLVRICFF-RECORD.
       06 FILLER                 PIC X(2).
       06 DELIVERY-STATUS        PIC X.


 FD  NEWORD
     LABEL RECORDS ARE STANDARD
     DATA RECORD IS NEWORD-RECORD.
 01  NEWORD-RECORD.
     COPY DDS-NORCD OF NEWORDLF.
 FD DISTRICT-FILE
      LABEL RECORDS ARE STANDARD.
 01 DISTRICT-RECORD.
      COPY DDS-DSRCD OF DSTRCT.
 FD  ORDERS
     LABEL RECORDS ARE STANDARD
     DATA RECORD IS ORDERS-RECORD.
 01  ORDERS-RECORD.
     COPY DDS-ORRCD OF ORDERSLF.
 FD  ORDERLINE
     LABEL RECORDS ARE STANDARD.
 01  ORDERLINE-RECORD.
     COPY DDS-OLRCD OF ORDLINLF.
 FD  CSTMR
     LABEL RECORDS ARE STANDARD
     DATA RECORD IS CSTMR-RECORD.
 01  CSTMR-RECORD.
     COPY DDS-CSRCD OF CSTMR.
 FD  DLVRYLOG
     LABEL RECORDS ARE STANDARD
     DATA RECORD IS DLVRYLOG-RECORD.
 01  DLVRYLOG-RECORD.
     COPY DDS-LGRCD OF DLVRYLOG.
     05 LOG-REDEF REDEFINES LGRCD.
         06       TIMEB    PIC S9(8).
         06       TIMEA    PIC S9(8).
         06       UNDEL    PIC S9.
         06       WID      PIC S9(4) COMP-4.

DFD  DBUG-FILE
D    LABEL RECORDS ARE STANDARD
D    DATA RECORD IS DBUG-REC2.
D   01  DBUGPRT-REC2.
D    COPY DDS-DBUGRCD2 OF DBUGPRT2.
D       05  DBUG-REC2 REDEFINES DBUGRCD2-O.
D           06  DEBUG-DATA.
D               07  ISONUMBER              PIC 99.
D               07  TXTDATA                PIC X(40).
D               07  WID                    PIC S9(4).
D               07  DID                    PIC S9(2).
D               07  TIME-DATE.
D                   08  DATE8              PIC 9(8).
D                   08  TIME6              PIC 9(6).
D               07  CID                    PIC S9(4).
D               07  CWID                   PIC S9(4).
D               07  CDID                   PIC S9(2).
D               07  CBAL                   PIC S9(11)V99.
D               07  PAYMNT                 PIC S9(7)V99.
DFD  ISO-IFILE
D    LABEL RECORDS ARE STANDARD.
D01  ISOIFILE-RECORD.
D    COPY DDS-ISOIRCD OF ISOIFILE.
D    05 ISO-IREC REDEFINES ISOIRCD.
D      06 ISOINUM PIC S9(4) COMP-4.



 WORKING-STORAGE SECTION.

 01 CSTMR-KEY.
   03 CSTMR-KEY-DATA OCCURS 5 TIMES.
      05  KNAM   PIC X(10).
      05  KVAL   PIC S9(9) COMP-4.

 01 CSTMR-HASH-PTR   USAGE POINTER.

 01 CSTMR-HASH-NAME  PIC X(10).
 01 FUNCT            PIC X(1).
 01 KEYS             PIC S9(9) COMP-4.
 01 RET-CODE         PIC S9(9) COMP-4.

 01  RET-CODE-PTR    USAGE POINTER.
 01  CSTMR-DEF-PTR   USAGE POINTER.
 01  CSTMR-KEY-PTR   USAGE POINTER.

D01  CBAL2              PIC S9(11)V99 COMP-3.
D01  ISOVALUE           PIC S999 COMP-3 VALUE 0.

 01 UPDATER               PIC X(1) VALUE "3".
 01 FETCHR                PIC X(1) VALUE "1".
 01 FETCHUPDATE           PIC X(1) VALUE "2".

 01 CUST-KEYS             pic 9(1) comp-4 vALUE 3.
 01 STOCK-KEYS            pic 9(1) comp-4 VALUE 2.
 01 ITEM-KEYS             pic 9(1) comp-4 VALUE 1.


  01   TRANSACTION-INPUT.
         06    TXN-TYPE          PIC X.
         06    JOBNAME           PIC X(10).
         06    CLIENT-INPUT      PIC X(202).
         06 DLVRY-I REDEFINES CLIENT-INPUT.
           08 WID                PIC S9(4) COMP-4.
           08 CARRIER            PIC XX.
           08 D-TIME-OF-DAY      PIC X(8).
*          08 D-DATE             PIC 9(6).

 77  STATUS-IND             PIC X(2).
 77  MAJ-MIN-SAV            PIC X(4).
 77  INDON                  PIC 1 VALUE B"1".
 77  INDOFF                 PIC 1 VALUE B"0".
 77  LEN                    PIC 9(10)V9(5) COMP.

 01 CMNF-INDIC-AREA.
    05  CMNF-INDIC          PIC 1 OCCURS 99 TIMES
                            INDICATOR 1.
```

```
 01 MAJ-MIN.
    05 MAJ                    PIC X(2).
    05 MIN                    PIC X(2).

*01  TPCDATE.
*   05 TPCDATEDD        PIC  99.
*   05 TPCDATEMM        PIC  99.
*   05 TPCDATECC        PIC  99.
*   05 TPCDATEYY        PIC  99.
*01  CURTIME.
*   05 CURTIME6        PIC 9(6).
*   05 FILLER          PIC 9(2).
 01  TIME-OF-DAY       PIC X(8).

D   01 TIME-OF-DAY-STRING PIC X(14).

D   01 TORD   PIC X(1) VALUE "b".

*   MI Time of day from call to mattod
*   05 DATEIYY          PIC  99.
*   05 DATEIMM          PIC  99.
*   05 DATEIDD          PIC  99.

 01 DQDATA-FRM-CLNT.
    05 DQDATA-WHS  PIC S9(4) COMP-4.
    05 DQDATA-CARR PIC X(2).
    05 DQDATA-BTIM PIC S9(8).
    05 DQDATA-fill PIC S9(6).

D  01 TESTNUM                  PIC 99 VALUE 1.

 01 OLINES-LEFT  PIC S9(2) COMP-4.

 77  ORDAMT PIC 9(9)V99.

 01  FILE-STATUS-ERROR-CDS.
    05  NEWORD-FILE-STATUS     PIC X(2).
 01  ORDERS-DELIVERED-TABLE.
       10 ORDER-LINE OCCURS 10 TIMES.
         15 ELEMENT-DISTRICT     PIC S99.
         15 ELEMENT-ORDER        PIC S9(6).
 77  WHS   PIC X(4).

 LINKAGE SECTION.

 PROCEDURE DIVISION.

 DELIVERY-PROGRAM.

      OPEN I-O NEWORD.
      OPEN I-O DISTRICT-FILE.
      OPEN I-O ORDERS.
      OPEN I-O CSTMR.
      OPEN I-O ORDERLINE.
      OPEN I-O DLVRICFF.
      OPEN EXTEND DLVRYLOG.
D     OPEN OUTPUT DBUG-FILE.
D     OPEN INPUT ISO-IFILE.

      MOVE "CSTMR"  TO CSTMR-HASH-NAME.
      MOVE "CID"    TO KNAM OF CSTMR-KEY-DATA(1).
      MOVE "CDID"   TO KNAM OF CSTMR-KEY-DATA(2).
      MOVE "CWID"   TO KNAM OF CSTMR-KEY-DATA(3).

      SET CSTMR-HASH-PTR TO ADDRESS OF CSTMR-HASH-NAME.
      SET RET-CODE-PTR   TO ADDRESS OF RET-CODE.
      SET CSTMR-DEF-PTR  TO ADDRESS OF CSTMR-RECORD.
      SET CSTMR-KEY-PTR  TO ADDRESS OF CSTMR-KEY.

 PROCESS-DELIVERY.

      READ DLVRICFF
      INTO TRANSACTION-INPUT INDICATORS ARE CMNF-INDIC-AREA.
      IF MAJ = "81"
          GO TO SHUTDOWN
      END-IF.

      MOVE DLVRY-I TO DQDATA-FRM-CLNT.

      MOVE D-TIME-OF-DAY OF DLVRY-I TO
                        LOGTIMEB OF DLVRYLOG-RECORD.

D     READ ISO-IFILE.
D     MOVE ISOINUM TO ISONUMBER, ISOVALUE.
      MOVE   0 TO NODID.
      MOVE   0 TO NOOID.
      MOVE "0" TO UNDELDST.
      MOVE DQDATA-WHS TO NOWID, OWID, OLWID, CWID OF CSTMR-RECORD.
      MOVE DQDATA-WHS TO DWID.

      IF DQDATA-CARR = "99" THEN
        PERFORM SHUTDOWN.

      MOVE DQDATA-CARR TO OCARID.


*     ACCEPT DATEINT     FROM DATE.
      CALL "gettime" USING  TIME-OF-DAY.
*     Move "19" to TPCDatecc.
*     If dateiyy < "70" then
*       Move "20" to TPCDatecc.
*     Move   DATEIYY      TO TPCDATEYY.
*     Move   DATEIMM      TO TPCDATEMM.
*     Move   DATEIDD    TO TPCDATEDD.

*     ACCEPT CURTIME FROM TIME.

* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D     MOVE "DEL PRE-READ  " TO TXTDATA.
D     MOVE 1 TO CDID OF DBUG-REC2.
D     MOVE DQDATA-WHS TO CWID OF DBUG-REC2.
D     MOVE 0 TO CID OF DBUG-REC2.
D     MOVE 1 TO DID OF DBUG-REC2.
D     MOVE 0 TO PAYMNT OF DBUG-REC2.
D     MOVE DQDATA-WHS TO WID OF DBUG-REC2.
D     MOVE 0 TO CBAL OF DBUG-REC2.
*     MOVE CURTIME6 TO TIME6.
*     MOVE TPCDATE TO DATE6.
D     MOVE "b" TO TORD.
D     CALL "getdttmstr" USING TIME-OF-DAY, TIME-OF-DAY-STRING,
D       BY VALUE TORD.
D     MOVE TIME-OF-DAY-STRING TO TIME-DATE OF DBUG-REC2.
D     WRITE DBUGPRT-REC2 FORMAT IS "DBUGRCD2".
      PERFORM 10 TIMES

        MOVE 0 TO ORDAMT
        ADD 1 TO NODID
        READ NEWORD
        IF NEWORD-FILE-STATUS = "23"
           MOVE NODID TO DID OF DISTRICT-RECORD
           READ DISTRICT-FILE
           READ NEWORD
           END-IF
D       IF NEWORD-FILE-STATUS = "23"
*          ACCEPT CURTIME FROM TIME
*          MOVE CURTIME6 TO TIME6
*          MOVE TPCDATE TO DATE6
D          CALL "gettime" USING TIME-OF-DAY
D          MOVE "b" TO TORD
D          CALL "getdttmstr" USING TIME-OF-DAY, TIME-OF-DAY-STRING,
D          BY VALUE TORD
D          MOVE TIME-OF-DAY-STRING TO TIME-DATE OF DBUG-REC2
D          MOVE NODID TO DID OF DBUG-REC2
D          MOVE "DEL NO NEWORDER" TO TXTDATA
D          WRITE DBUGPRT-REC2 FORMAT IS "DBUGRCD2"
D          CALL "delayme"
D          READ NEWORD
*          ACCEPT CURTIME FROM TIME
*          MOVE CURTIME6 TO TIME6
*          MOVE TPCDATE TO DATE6
D          CALL "gettime" USING TIME-OF-DAY
D          MOVE "b" TO TORD
D          CALL "getdttmstr" USING TIME-OF-DAY, TIME-OF-DAY-STRING,
D          BY VALUE TORD
D          MOVE TIME-OF-DAY-STRING TO TIME-DATE OF DBUG-REC2
D          MOVE "DEL NO NEWORD2" TO TXTDATA
D          WRITE DBUGPRT-REC2 FORMAT IS "DBUGRCD2"
D          END-IF
           IF NEWORD-FILE-STATUS = "23"
             MOVE "1" TO UNDELDST
           ELSE
             MOVE NODID TO ODID, OLDID, CDID OF CSTMR-RECORD
             MOVE NOOID TO OID, OLOID

             READ ORDERS
             MOVE DQDATA-CARR TO OCARID
             REWRITE ORDERS-RECORD

             READ ORDERLINE
*            MOVE TPCDATE TO OLDLVD
*            MOVE CURTIME6 TO OLDLVT
             MOVE TIME-OF-DAY TO OLDLVTOD
             REWRITE ORDERLINE-RECORD
             ADD OLAMNT TO ORDAMT

             SUBTRACT 1 FROM OLINES GIVING OLINES-LEFT

             PERFORM OLINES-LEFT TIMES
                  READ ORDERLINE NEXT
*                 MOVE TPCDATE TO OLDLVD
*                 MOVE CURTIME6 TO OLDLVT
                  MOVE TIME-OF-DAY TO OLDLVTOD
                  REWRITE ORDERLINE-RECORD
                  ADD OLAMNT TO ORDAMT
             END-PERFORM


           MOVE OCID TO CID OF CSTMR-RECORD
           MOVE CID  OF CSTMR-RECORD TO KVAL OF CSTMR-KEY-DATA(1)
           MOVE CDID OF CSTMR-RECORD TO KVAL OF CSTMR-KEY-DATA(2)
           MOVE CWID OF CSTMR-RECORD TO KVAL OF CSTMR-KEY-DATA(3)

             CALL "qdbrunha" USING BY VALUE
             CSTMR-HASH-PTR FETCHUPDATE CUST-KEYS CSTMR-KEY-PTR
                    CSTMR-DEF-PTR RET-CODE-PTR
             ADD ORDAMT TO CBAL OF CSTMR-RECORD
             ADD 1 TO CDELCNT

             CALL "qdbrunha" USING BY VALUE
                  CSTMR-HASH-PTR UPDATER CUST-KEYS CSTMR-KEY-PTR
                       CSTMR-DEF-PTR RET-CODE-PTR

             DELETE NEWORD
             MOVE OID  OF ORDERS-RECORD
                    TO ELEMENT-ORDER(NODID)

         END-IF
      END-PERFORM.

* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D     IF ISOVALUE = 8 THEN GO TO DBG-SKP1.
D     IF ISOVALUE = 6 THEN
D        MOVE "DEL PRE-ROLLBK" TO TXTDATA
D     ELSE IF ISOVALUE = 5 THEN
D        MOVE "DEL PRE-COMMIT" TO TXTDATA.
D     MOVE 1 TO CDID OF DBUG-REC2.
D     MOVE DQDATA-WHS TO CWID OF DBUG-REC2.
D     MOVE 1 TO DID OF DBUG-REC2.
D     MOVE DQDATA-WHS TO WID OF DBUG-REC2.
*     MOVE CURTIME6 TO TIME6.
*     MOVE TPCDATE TO DATE6.
D     MOVE "b" TO TORD.
D     CALL "getdttmstr" USING TIME-OF-DAY, TIME-OF-DAY-STRING,
D       BY VALUE TORD.
D     MOVE TIME-OF-DAY-STRING TO TIME-DATE OF DBUG-REC2.
D     MOVE TIME-OF-DAY-STRING TO TIME-DATE OF DBUG-REC2.
D     WRITE DBUGPRT-REC2 FORMAT IS "DBUGRCD2".
D     CALL "delayme".
D     IF ISOVALUE = 6 THEN
D      ROLLBACK.
DDBG-SKP1.
              COMMIT.

* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D     IF ISOVALUE = 8 THEN GO TO DBG-SKP2.
D     IF ISOVALUE = 6 THEN
D        MOVE "DEL POST-ROLLBK" TO TXTDATA
D     ELSE IF ISOVALUE = 5 THEN
D        MOVE "DEL POST-COMMIT" TO TXTDATA.
D     MOVE 1 TO CDID OF DBUG-REC2.
D     MOVE DQDATA-WHS TO CWID OF DBUG-REC2.
D     MOVE 1 TO DID OF DBUG-REC2.
D     MOVE 0 TO PAYMNT OF DBUG-REC2.
D     MOVE DQDATA-WHS TO WID OF DBUG-REC2.
D     MOVE CBAL2 TO CBAL OF DBUG-REC2.
*     ACCEPT DATEINT     FROM DATE.
*     Move "19" to TPCDatecc.
*     If dateiyy < "70" then
*       Move "20" to TPCDatecc.
*     Move   DATEIYY      TO TPCDateYY.
*     Move   DATEIMM      TO TPCDateMM.
*     Move   DATEIDD      TO TPCDateDD.
*     ACCEPT CURTIME FROM TIME.
*     MOVE CURTIME6 TO TIME6.
*     MOVE TPCDATE TO DATE6.
D     CALL "gettime" USING TIME-OF-DAY.
D     MOVE "b" TO TORD.
D     CALL "getdttmstr" USING TIME-OF-DAY, TIME-OF-DAY-STRING,
D          BY VALUE TORD.
D     MOVE TIME-OF-DAY-STRING TO TIME-DATE OF DBUG-REC2.
D     WRITE DBUGPRT-REC2 FORMAT IS "DBUGRCD2".
DDBG-SKP2.
```

```
           MOVE DQDATA-WHS TO WH OF DLVRYLOG-RECORD.
      *    ACCEPT LOGTIMEA FROM TIME.
           CALL "gettime" USING TIME-OF-DAY.
           MOVE TIME-OF-DAY TO LOGTIMEA.
           WRITE DLVRYLOG-RECORD.

           GO TO PROCESS-DELIVERY.

           EXIT.

       SHUTDOWN.
           CLOSE ORDERS
                 ORDERLINE
                 NEWORD
                 CSTMR
                 DISTRICT-FILE
                 DLVRYLOG
                 DLVRICFF.
       D   CLOSE DBUG-FILE.
       D   CLOSE ISO-IFILE.
           STOP RUN.
```

ATPCCMTRA: New Order Start Program

```
 /* This program:  Begins commitment control                    +
                   Retrieves the warehouse id and district id    +
                     for each user                             */  +

             PGM

             DCL        VAR(&SSNWHSE) TYPE(*CHAR) LEN(4)

             DCL        VAR(&DATALIB) TYPE(*CHAR) LEN(10) +
                          VALUE(TPCCD697)
             DCL        VAR(&OBJLIB) TYPE(*CHAR) LEN(10) +
                          VALUE(TPCCNCSRVR)
             DCL        VAR(&QNAME) TYPE(*CHAR) LEN(10) VALUE(INITDTAQ)
             DCL        VAR(&QLIB) TYPE(*CHAR) LEN(10)
             DCL        VAR(&FLDLEN) TYPE(*DEC) LEN(5 0) VALUE(6)
             DCL        VAR(&SSNDSTRCT) TYPE(*CHAR) LEN(2)
             DCL        VAR(&VALUE) TYPE(*CHAR) LEN(6)

             MONMSG     MSGID(CPF0000)

             ADDLIBLE   LIB(&DATALIB)
             CHGJOB     RUNPTY(10)
             OVRDBF     FILE(ORDLIN) TOFILE(*FILE) SHARE(*NO) +
                          SEQONLY(*YES 16)
             OVRDBF     FILE(NEWORD) TOFILE(*FILE) SHARE(*NO) +
                          SEQONLY(*YES 2)
             OVRDBF     FILE(ORDLINLF) TOFILE(*FILE) NBRRCDS(15) +
                          SHARE(*NO)
             OVRDBF     FILE(HSTRY) TOFILE(*FILE) SHARE(*NO) +
                          SEQONLY(*NO)
             OVRDBF     FILE(ORDERS) TOFILE(*FILE) WAITRCD(*NOMAX) +
                          SHARE(*NO) SEQONLY(*YES 2)
             OVRDBF     FILE(ORDERSLF) TOFILE(*FILE) WAITRCD(*NOMAX) +
                          SHARE(*NO) SEQONLY(*YES 2)
             OVRDBF     FILE(STOCKPF) TOFILE(*FILE) WAITRCD(2) +
                          SHARE(*NO)
             OVRDBF     FILE(STOCK) TOFILE(*FILE) WAITRCD(2) SHARE(*NO)
             OVRDBF     FILE(CSTMRPF) TOFILE(*FILE) WAITRCD(5) +
                          SHARE(*NO)
             OVRDBF     FILE(CSTMR) TOFILE(*FILE) WAITRCD(5) SHARE(*NO)
             OVRDBF     FILE(DSTRCT) TOFILE(*FILE) SHARE(*NO)
             OVRDBF     FILE(CSTMRLFNAM) SHARE(*NO) NBRRCDS(5)
             OVRDBF     FILE(CSTMRLFCRT) SHARE(*NO)
             OVRDBF     FILE(WRHS) TOFILE(WRHS) LVLCHK(*NO) +
                          SHARE(*NO)
             STRCMTCTL  LCKLVL(*ALL)
             MONMSG     MSGID(CPF0000)
 FIRSTCALL:  CALL       PGM(&OBJLIB/NOPAYOSMTR)


ENDIT:       ENDPGM
```

NOPAYOSMTR: Neworder, Payment, Order, Status Transactions Program

```
        PROCESS NOTRUNC NORANGE.
       IDENTIFICATION DIVISION.
         PROGRAM-ID. NOPAYOSMTR.
         AUTHOR. P/P COC
         INSTALLATION. IBM-ROCHESTER.
         DATE-WRITTEN.
         DATE-COMPILED.
      *                                                               *
      *   C H A N G E S    &     U P D A T E S
      *
      *    LATEST CHANGES LISTED FIRST
      *
      *    MM/DD/YY   AUTHOR NAME          LINES CHANGED/ADDED: NNN
      *
      *
      *
      *                                                               *
       ENVIRONMENT DIVISION.
       CONFIGURATION SECTION.
         SOURCE-COMPUTER. IBM-AS400.
      *  SOURCE-COMPUTER. IBM-AS400 WITH DEBUGGING MODE.
         OBJECT-COMPUTER. IBM-AS400.
         INPUT-OUTPUT SECTION.
         FILE-CONTROL.
             SELECT MAINICFF ASSIGN TO WORKSTATION-MAINICFF-SI
                    ORGANIZATION IS TRANSACTION
                    FILE STATUS IS STATUS-IND MAJ-MIN.
             SELECT WAREHOUSE-FILE ASSIGN TO DATABASE-WRHS
                  ORGANIZATION IS INDEXED
                  ACCESS MODE IS RANDOM
                  RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
                  WITH DUPLICATES.
             SELECT DISTRICT-FILE ASSIGN TO DATABASE-DSTRCT
                  ORGANIZATION IS INDEXED
                  ACCESS MODE IS RANDOM
                  RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
                  WITH DUPLICATES
                  FILE STATUS IS DISTRICT-FILE-STATUS.
             SELECT CUSTOMER-FILE ASSIGN TO DATABASE-CSTMR
                  ORGANIZATION IS INDEXED
                  ACCESS MODE IS RANDOM
                  RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
                  WITH DUPLICATES
                  FILE STATUS IS CUSTOMER-FILE-STATUS.
             SELECT ITEM-FILE ASSIGN TO DATABASE-ITEM
                  ORGANIZATION IS INDEXED
                  ACCESS MODE IS RANDOM
                  RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
                  WITH DUPLICATES
                  FILE STATUS IS ITEM-FILE-STATUS.
             SELECT STOCK-FILE ASSIGN TO DATABASE-STOCK
                  ORGANIZATION IS INDEXED
                  ACCESS MODE IS RANDOM
                  RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
                  WITH DUPLICATES
                  FILE STATUS IS STOCK-FILE-STATUS.
             SELECT NEWORDER-FILE ASSIGN TO DATABASE-NEWORD
                  ORGANIZATION IS SEQUENTIAL
                  FILE STATUS IS NEWORDER-FILE-STATUS.
             SELECT ORDERS-FILE ASSIGN TO DATABASE-ORDERSLF
                  ORGANIZATION IS SEQUENTIAL.
             SELECT ORDERS-VIEW ASSIGN TO DATABASE-ORDERS
                  ORGANIZATION IS INDEXED
                  ACCESS MODE IS DYNAMIC
                  RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
                  WITH DUPLICATES.
             SELECT ORDERLINE-FILE ASSIGN TO DATABASE-ORDLIN
                  ACCESS MODE IS SEQUENTIAL.
             SELECT ORDERLINE-VIEW ASSIGN TO DATABASE-ORDLINLF
                  ORGANIZATION IS INDEXED
                  ACCESS MODE IS DYNAMIC
                  RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
                  WITH DUPLICATES.
             SELECT HISTORY-FILE ASSIGN TO DATABASE-HSTRY
                  ORGANIZATION IS SEQUENTIAL.
       D     SELECT DBUG-FILE
       D          ASSIGN TO FORMATFILE-DBUGPRT
       D          ORGANIZATION IS SEQUENTIAL
       D          ACCESS IS SEQUENTIAL.
       D     SELECT DBUG-FILE2
       D          ASSIGN TO FORMATFILE-DBUGPRT2
       D          ORGANIZATION IS SEQUENTIAL
       D          ACCESS IS SEQUENTIAL.
       D     SELECT ISO-IFILE
       D          ASSIGN TO DATABASE-ISOIFILE
       D          ORGANIZATION IS SEQUENTIAL
       D          ACCESS IS SEQUENTIAL.
             SELECT CUST-REL-FILE ASSIGN TO DATABASE-CSTMRPF
                  ORGANIZATION IS RELATIVE
                  ACCESS MODE IS RANDOM RELATIVE KEY IS CUSTREL
                  FILE STATUS IS CUSTOMER-FILE-STATUS.

         I-O-CONTROL.
           COMMITMENT CONTROL FOR WAREHOUSE-FILE
           COMMITMENT CONTROL FOR DISTRICT-FILE
           COMMITMENT CONTROL FOR CUSTOMER-FILE
           COMMITMENT CONTROL FOR CUST-REL-FILE
           COMMITMENT CONTROL FOR STOCK-FILE
           COMMITMENT CONTROL FOR ITEM-FILE
           COMMITMENT CONTROL FOR NEWORDER-FILE
           COMMITMENT CONTROL FOR ORDERS-FILE
           COMMITMENT CONTROL FOR ORDERS-VIEW
           COMMITMENT CONTROL FOR ORDERLINE-FILE
           COMMITMENT CONTROL FOR ORDERLINE-VIEW
           COMMITMENT CONTROL FOR HISTORY-FILE.
       DATA DIVISION.
       FILE SECTION.
       FD MAINICFF
         LABEL RECORDS ARE STANDARD.
       01  ICFREC.
         COPY DDS-ALL-FORMATS OF MAINICFF.

          05 NEWORD-O REDEFINES MAINICFF-RECORD.
               06    NEWORD-RESULT          PIC X.
               06    OENTTM                 PIC S9(6).
               06    CLAST                  PIC X(16).
               06    CCREDT                 PIC X(2).
               06    CDCT                   PIC S999V99 COMP-3.
               06    OID                    PIC S9(9) COMP-4.
               06    WTAX                   PIC S9V9999 COMP-3.
               06    DTAX                   PIC S9V9999 COMP-3.
               06    TOTAMT                 PIC S9(9)V9(2) COMP-3.
               06  OUTPUT-TABLE.
                 10 OUTPUT-LINE OCCURS  16 TIMES.
                    15 OUTPUT-STQTY        PIC S9(4) COMP-4.
                    15 OUTPUT-BORG         PIC  X(1).
                    15 OUTPUT-ITEM-INFO.
                      20 OUTPUT-INAME        PIC X(24).
                      20 OUTPUT-IPRICE       PIC S9(3)V9(2) COMP-3.
                    15 OUTPUT-OLAMNT       PIC S9(5)V9(2) COMP-3.

            05 PAYMNT-O REDEFINES MAINICFF-RECORD.
               06 OUTPUT-FMT-NUM-3      PIC XX.
      *                                    17 = bad credit
      *                                    18 = no cid or clast
      *                                    19 = good credit
      *                                    88 = bad last name
               06 HDT.
                 08 HOTOD    PIC X(8).
                 08 HTIME    pic 9(6).
               06 WH-DATA.
                 08 WADDR1   PIC X(20).
                 08 WADDR2   PIC X(20).
                 08 WCITY    PIC X(20).
                 08 WSTATE   PIC X(02).
                 08 WZIP     PIC X(9).
               06 DST-DATA.
                 08 DADDR1   PIC X(20).
                 08 DADDR2   PIC X(20).
                 08 DCITY    PIC X(20).
                 08 DSTATE   PIC X(02).
                 08 DZIP     PIC X(9).
               06 CID          PIC S9(9) COMP-4.
               06 PAYMENT-CUST-INFO.
                  08 CFIRST   PIC X(16).
                  08 CINIT    PIC X(02).
                  08 CLAST    PIC X(16).
                  08 CLDATE   PIC S9(8).
                  08 CADDR1   PIC X(20).
                  08 CCREDT   PIC X(02).
                  08 CADDR2   PIC X(20).
                  08 CDCT     PIC S999V99 COMP-3.
                  08 CCITY    PIC X(20).
                  08 CSTATE   PIC X(02).
                  08 CZIP     PIC X(9).
```

```
          08 CPHONE   PIC X(16).
          08 CBAL     PIC S9(11)V9(2) COMP-3.
          08 CCRDLM   PIC S9(11)V9(2) COMP-3.
       06 MISC-CDATA.
          08  CDAT1    PIC X(50).
          08  CDAT2    PIC X(50).
          08  CDAT3    PIC X(50).
          08  CDAT4    PIC X(50).


    05  ORDSTS-O REDEFINES MAINICFF-RECORD.
        06 ORDSTS-FMT-NUM        PIC XX.
        06 OCID                  PIC S9(9) COMP-4.
        06 CUSTINFO.
           08 CFIRST               PIC X(16).
           08 CINIT                PIC X(2).
           08 CLAST                PIC X(16).
        06 CBAL                  PIC S9(11)V9(2) COMP-3.
        06 ORDERINFO.
           08 OID                   PIC S9(9) COMP-4.
           08 ODT.
              10 OENTTOD               PIC X(8).
              10 OENTTM                PIC 9(6).
           08 OCARID                PIC X(2).
           08 OLINENBR              PIC S9(4) COMP-4.
        06 OLINEINFO OCCURS 15 TIMES.
          07 DO-OLSPWH           PIC S9(4) COMP-4.
          07 DO-OLIID            PIC S9(9) COMP-4.
          07 DO-OLQTY            PIC S9(3) COMP-3.
          07 DO-OLAMNT           PIC S9(5)V9(2) COMP-3.
*         07 DO-OLDLVD           PIC S9(8).
          07 DO-OLDLVTOD         PIC X(8).
    05 STKLVL-O REDEFINES MAINICFF-RECORD.
        06 FILLER                PIC X(2).
        06 BLWSTK                PIC S9(4) COMP-4.
    05 DLVRY-O REDEFINES MAINICFF-RECORD.
        06 FILLER                PIC X(2).
        06 DELIVERY-STATUS       PIC X.


   FD WAREHOUSE-FILE
      LABEL RECORDS ARE STANDARD.
    01 WAREHOUSE-REC.
         COPY DDS-WRRCD OF WRHS.
         05 WRHS-INFO REDEFINES WRRCD.
            06 FILLER             PIC X(12).
            06 WH-ADDRESS.
               08 WADDR1           PIC X(20).
               08 WADDR2           PIC X(20).
               08 CITYW            PIC X(20).
               08 STATEW           PIC X(2).
               08 ZIPW             PIC X(10).
   FD DISTRICT-FILE
      LABEL RECORDS ARE STANDARD.
    01 DISTRICT-REC.
      COPY DDS-DSRCD OF DSTRCT.
       05  DISTRICT-INFO REDEFINES DSRCD.
           06 DIST-KEY.
              08 DID              PIC S9(4) COMP-4.
              08 DWID             PIC S9(4) COMP-4.
           06 FILLER             PIC X(10).
           06 DIST-ADDRESS.
              08 DADDR1            PIC X(20).
              08 DADDR2            PIC X(20).
              08 CITY              PIC X(20).
              08 STATE             PIC X(2).
              08 ZIP               PIC X(10).
   FD  CUSTOMER-FILE
       LABEL RECORDS ARE STANDARD.
    01  CUSTOMER-REC.
          COPY DDS-CSRCD OF CSTMR.
         05  CUST-KEY REDEFINES CSRCD.
             06 CIDK     PIC S9(9) COMP-4.
             06 CDIDK    PIC S9(4) COMP-4.
             06 CWIDK    PIC S9(4) COMP-4.
         05  CUST-INFO-REC REDEFINES CSRCD.
             06 FILLER   PIC X(8).
             06 CUST-INFO.
                08 CFIRSTP  PIC X(16).
                08 CINITP   PIC X(02).
                08 CLASTP   PIC X(16).
                08 CLDATEP  PIC S9(8).
                08 CADDR1P  PIC X(20).
                08 CCREDTP  PIC X(02).
                08 CADDR2P  PIC X(20).
                08 CDCTP    PIC S9(1)V9(4) COMP-3.
                08 CCITYP   PIC X(20).
                08 CSTATEP  PIC X(02).
                08 CZIPP    PIC X(10).
                08 CPHONEP  PIC X(16).
                08 CBAL     PIC S9(11)V999 COMP-3.
                08 CCRDLM   PIC S9(11)V999 COMP-3.

   FD  CUST-REL-FILE
       LABEL RECORDS ARE STANDARD.
    01  CUST-REL-REC.
          COPY DDS-CSRCD OF CSTMRPF.
   FD ITEM-FILE
      LABEL RECORDS ARE STANDARD.
    01 ITEM-REC.
      COPY DDS-ITRCD OF ITEM.
      05 ITEM-DEF REDEFINES ITRCD.
          06 ID-IID            PIC S9(9) COMP-4.
          06 ID-IMID           PIC S9(9) COMP-4.
          06 ITEM-INFO.
             08 ID-INAME       PIC X(24).
             08 ID-IPRICE      PIC S9(3)V99 COMP-3.
          06 ID-IDATA          PIC X(50).
   FD STOCK-FILE
      LABEL RECORDS ARE STANDARD.
    01 STOCK-REC.
      COPY DDS-STRCD OF STOCK.
      05  STOCK-DEF REDEFINES STRCD.
          06  STOCK-IN.
              08 STWID1          PIC S9(4) COMP-4.
              08 STIID1          PIC S9(9) COMP-4.
          06 FILLER              PIC S9(4) COMP-4.
          06  DIST-INFO  OCCURS 10 TIMES.
              07  DIST-IN        PIC X(24).
   FD NEWORDER-FILE
      LABEL RECORDS ARE STANDARD.
    01 NEWORDER-REC.
      COPY DDS-NORCD OF NEWORD.
   FD ORDERS-FILE
      LABEL RECORDS ARE STANDARD.
    01 ORDERS-REC.
      COPY DDS-ORRCD OF ORDERSLF.
   FD ORDERS-VIEW
      LABEL RECORDS ARE STANDARD.
    01 ORDERS-VIEW-REC.
      COPY DDS-ORRCD OF ORDERS.
   FD ORDERLINE-FILE
      LABEL RECORDS ARE STANDARD.
    01 ORDERLINE-REC.
      COPY DDS-OLRCD OF ORDLIN.
      05 ORDERLIN-INPUT REDEFINES OLRCD.
         07 ORDER-KEY.
            08 OLOID                 PIC S9(9) COMP-4.
            08 OLDID                 PIC S9(4) COMP-4.
            08 OLWID                 PIC S9(4) COMP-4.
            08 OLNBR                 PIC S9(4) COMP-4.
         07 OL-INPUT.
            08 OL-INPUT-OLSPWH       PIC S9(4) COMP-4.
            08 OL-INPUT-OLIID        PIC S9(9) COMP-4.
            08 OL-INPUT-QTY          PIC S9(3) COMP-3.
         07 FILLER                   PIC X(4).
*        07 OLDATE-TIME              PIC X(14).
         07 OLTIME-OF-DAY            PIC X(8).
          07 TIME-FILLER             PIC X(6).
   FD ORDERLINE-VIEW
      LABEL RECORDS ARE STANDARD.
    01 ORDERLINE-VIEW-REC.
      COPY DDS-OLRCD OF ORDLINLF.
      05 OL-STATUS-OUTPUT REDEFINES OLRCD.
         08 FILLER            PIC X(10).
         08 OL-INFO.
            10 OL-INFO-OLSPWH       PIC S9(4) COMP-4.
            10 OL-INFO-OLIID        PIC S9(9) COMP-4.
            10 OL-INFO-OLQTY        PIC S9(3) COMP-3.
            10 OL-INFO-OLAMNT       PIC S9(5)V9(2) COMP-3.
*           10 OL-INFO-OLDLVD       PIC S9(8).
            10 OL-INFO-OLDLVTOD     PIC X(8).
    FD  HISTORY-FILE
        LABEL RECORDS ARE STANDARD
        DATA RECORD IS HISTORY-REC.
        01  HISTORY-REC.
         COPY DDS-HSRCD OF HSTRY.
           05 HIST-INFO REDEFINES HSRCD.
              07 HISTORY-DATA.
                09 PAY-DIST-KEY.
                 10 DID               PIC S9(4) COMP-4.
                 10 WID               PIC S9(4) COMP-4.
                09 PAY-CUST-KEY.
                 10 CID               PIC S9(9) COMP-4.
                 10 CDID              PIC S9(4) COMP-4.
                 10 CWID              PIC S9(4) COMP-4.
              07 DATE-TIME.
                09 HTOD               PIC X(8).
*               09 HTIME              PIC 9(6).
              07 AMOUNT             PIC S9(5)V99 COMP-3.
              07 HDATA.
                09 WNAME              PIC X(10).
                09 FILLER             PIC XXXX.
                09 DNAME              PIC X(10).
   DFD  DBUG-FILE
   D    LABEL RECORDS ARE STANDARD
   D    DATA RECORD IS DBUG-REC.
   D   01  DBUGPRT-REC.
   D    COPY DDS-DBUGRCD OF DBUGPRT.
   D       05  DBUG-REC REDEFINES DBUGRCD-O.
   D           06  DEBUG-DATA.
   D               07  ISONUM                 PIC XX.
   D               07  TXTDATA                PIC X(40).
   D               07  WID                    PIC S9(4).
   D               07  DID                    PIC S9(4).
   D               07  DATE6                  PIC S9(8).
   D               07  TIME6                  PIC S9(6).
   D               07  CID                    PIC S9(4).
   D               07  OID                    PIC S9(9).
   D               07  DEBUG-ITEM-INFO OCCURS 15 TIMES.
   D                 08 DEBUG-OLSPWH           PIC S9(4).
   D                 08 DEBUG-OLIID            PIC S9(6).
   D                 08 DEBUG-OLQTY            PIC S9(3).
   D                 08 DEBUG-IPRICE           PIC S9(5)V9(2).
   DFD  DBUG-FILE2
   D    LABEL RECORDS ARE STANDARD
   D    DATA RECORD IS DBUG-REC2.
   D   01  DBUGPRT-REC2.
   D    COPY DDS-DBUGRCD2 OF DBUGPRT2.
   D       05  DBUG-REC2 REDEFINES DBUGRCD2-O.
   D           06  DEBUG-DATA.
   D               07  ISONUM                 PIC XX.
   D               07  TXTDATA                PIC X(40).
   D               07  WID                    PIC S9(4).
   D               07  DID                    PIC S9(4).
   D               07  DATE6                  PIC S9(8).
   D               07  TIME6                  PIC S9(6).
   D               07  CID                    PIC S9(4).
   D               07  CWID                   PIC S9(4).
   D               07  CDID                   PIC S9(4).
   D               07  CBAL                   PIC S9(11)V99.
   D               07  PAYMNT                 PIC S9(7)V99.
   DFD  ISO-IFILE
   D    LABEL RECORDS ARE STANDARD.
   D01  ISOIFILE-RECORD.
   D    COPY DDS-ISOIRCD OF ISOIFILE.
   D    05 ISO-IREC REDEFINES ISOIRCD.
   D      06 ISOINUM PIC S9(4) COMP-4.


    WORKING-STORAGE SECTION.

    01 MAJ-MIN.
       03  MAJ    PIC X(2).
       03  MIN    PIC X(2).

    01 ITEM-KEY-PTR  USAGE POINTER.
    01 STOCK-KEY-PTR USAGE POINTER.
    01 CSTMR-KEY-PTR USAGE POINTER.

    01 ITEM-KEY.
      03 ITEM-KEY-DATA OCCURS 5 TIMES.
         05  KNAM   PIC X(10).
         05  KVAL   PIC S9(9) COMP-4.

    01 STOCK-KEY.
      03 STOCK-KEY-DATA OCCURS 5 TIMES.
         05  KNAM   PIC X(10).
         05  KVAL   PIC S9(9) COMP-4.

    01 CSTMR-KEY.
      03 CSTMR-KEY-DATA OCCURS 5 TIMES.
         05  KNAM   PIC X(10).
         05  KVAL   PIC S9(9) COMP-4.

    01 ITEM-HASH-NAME   PIC X(10).
    01 STOCK-HASH-NAME  PIC X(10).
    01 CSTMR-HASH-NAME  PIC X(10).

    01 ITEM-HASH-PTR    USAGE POINTER.
    01 STOCK-HASH-PTR   USAGE POINTER.
    01 CSTMR-HASH-PTR   USAGE POINTER.

    01 FUNCT            PIC X(1).
    01 KEYS             PIC S9(9) COMP-4.
    01 RET-CODE         PIC S9(9) COMP-4.

    01  RET-CODE-PTR    USAGE POINTER.
```

```
 01  ITEM-DEF-PTR    USAGE POINTER.
 01  STOCK-DEF-PTR   USAGE POINTER.
 01  CSTMR-DEF-PTR   USAGE POINTER.

 01 UPDATER                PIC X(1) VALUE "3".
 01 FETCHR                 PIC X(1) VALUE "1".
 01 FETCHUPDATE            PIC X(1) VALUE "2".

 01 CUST-KEYS              PIC 9(1) comp-4 VALUE 3.
 01 STOCK-KEYS             PIC 9(1) comp-4 VALUE 2.
 01 ITEM-KEYS              PIC 9(1) comp-4 VALUE 1.

 01 CMNF-INDIC-AREA.
    03 CMNF-INDIC     PIC 1 OCCURS 99 TIMES
                      INDICATOR 1.

  01   TRANSACTION-INPUT.
         06    TXN-TYPE           PIC X.
         06    JOBNAME            PIC X(10).
         06    CLIENT-INPUT       PIC X(202).
         06 NEWORD-I REDEFINES CLIENT-INPUT.
           08 CWID               PIC S9(4) COMP-4.
           08 CDID               PIC S9(4) COMP-4.
           08 CID                PIC S9(9) COMP-4.
           08 NUMBER-OF-ITEMS    PIC 9(3) COMP-3.
           08 INPUT-LINE OCCURS 15 TIMES.
            10 OLSPWH            PIC S9(4) COMP-4.
            10 OLIID             PIC S9(9) COMP-4.
            10 OLQTY             PIC S9(3) COMP-3.
         06 PAYMENT-I REDEFINES CLIENT-INPUT.
           08 PAYMENT-TYPE       PIC X.
*                               Payment type = C if by CID
*                               Payment type = L if by CLAST
           08 HISTORY-DATA.
            09 PAY-DIST-KEY.
             10 DID              PIC S9(4) COMP-4.
             10 WID              PIC S9(4) COMP-4.
            09 PAY-CUST-KEY.
             10 CID              PIC S9(9) COMP-4.
             10 CDID             PIC S9(4) COMP-4.
             10 CWID             PIC S9(4) COMP-4.
           08 AMOUNT             PIC S9(5)V99 COMP-3.
           08 CLAST              PIC X(16).
           08 FILLER             PIC X(155).
         06 ORDSTS-I REDEFINES CLIENT-INPUT.
           08 ORDSTS-TYPE        PIC X.
           08 ORD-CUST-KEY.
            09 CID               PIC S9(9) COMP-4.
            09 DID               PIC S9(4) COMP-4.
            09 WID               PIC S9(4) COMP-4.
           08 CLAST              PIC X(16).
           08 FILLER             PIC X(165).
         06 STKLVL-I REDEFINES CLIENT-INPUT.
           08 WID                PIC S9(4) COMP-4.
           08 DID                PIC S9(4) COMP-4.
           08 THRESHOLD          PIC S9(4) COMP-4.
           08 BLWSTK             PIC S9(4) COMP-4.
         06 DLVRY-I REDEFINES CLIENT-INPUT.
           08 WID                PIC S9(4) COMP-4.
           08 CARRIER            PIC XX.
           08 D-DATE             PIC 9(8).
           08 D-TIME             PIC 9(8).

 77 STATUS-IND PIC X(2).


  01 CWIDR    PIC S9(4) COMP-4.
  01 CDIDR    PIC S9(4) COMP-4.
  01 CLASTR   PIC X(16).
  01 CUSTREL      PIC 9(9) COMP-4.


 01 INPUT-ROW.
      05 STOCK-KEY-INPUT.
            10 OLSPWH            PIC S9(4) COMP-4.
            10 OLIID             PIC S9(9) COMP-4.
      05 OLQTY             PIC S9(3) COMP-3.

 01  CHAR-HIST-DATA.
     05 DID                  PIC S99.
     05 WID                  PIC S9(4).
     05 CID                  PIC S9(6).
     05 CDID                 PIC S99.
     05 CWID                 PIC S9(4).
     05 AMOUNT               PIC S9(5)V99.


   01 ROLL-BACK-REQUIRED   PIC X(1).
   01 CMD1 PIC X(14) VALUE "DLYJOB DLY(30)".
   01 CMD-LEN  PIC S9(10)V9(5) COMP-3 VALUE 14.
   01 CONTR-1                 PIC S999 COMP-4.

D  01 TESTNUM                 PIC 99 VALUE 1.
D  01 J                       PIC S999 COMP-3.

   01 DATEINT.
      06  YY                 PIC 9(2).
      06  MMDD               PIC 9(4).

*  01 DATETIME.
*     05 TPCDATE.
*       06  TPCMD             PIC 9(4).
*       06  TPCCEN            PIC 9(2)  VALUE 19.
*       06  TPCYEAR           PIC 9(2).
*     05 CURTIME.
*       06 CURTIME6        PIC 9(6).
*       06 FILLER          PIC 9(2).
   01 DATETIME-INIT.
      05 TPCDATE-INIT          PIC X(8) VALUE 0.
*     05 TPCTIME-INIT          PIC 9(6) VALUE 0.
*  01 DATETIME-CHARINIT REDEFINES DATETIME-INIT PIC X(14).
   01 DATETIME-CHARINIT REDEFINES DATETIME-INIT PIC X(8) VALUE 0.
   01 TIME-OF-DAY PIC X(8).

    01 CNTR                    PIC 99.
    01 CUST-INDEX   PIC S9(4)  USAGE BINARY.
    01 CUSTR-INDEX   PIC S9(4)  USAGE BINARY.
    01 CUSTOMER-ARRAY.
       03  CUSTOMER-ARRAY-ELEMENT   OCCURS 100 TIMES.
           05 CID      PIC S9(9) COMP-4.
           05 CDID     PIC S9(4) COMP-4.
           05 CWID     PIC S9(4) COMP-4.
   01 TEMP-DATA2        PIC X(468).

   01 HOST-DATA.
       03  HOST-ARRAY-ELEMENT   OCCURS 100 TIMES.
           05 RRN    PIC S9(9) COMP-4.

   01 ERROR-HDLNG-PARAMETERS.
       05  NEWORDER-FILE-STATUS           PIC X(2).
       05  ITEM-FILE-STATUS               PIC X(2).
       05  STOCK-FILE-STATUS              PIC X(2).
       05  CUSTOMER-STATUS-2              PIC X(2).
       05  DISTRICT-FILE-STATUS           PIC X(2).
       05  CUSTOMER-FILE-STATUS           PIC X(2).

  01  SWITCH-AREA.
      05  SW03                            PIC 1.
        88 LOCK-OCCURRED               VALUE B"1".
        88 LOCK-OFF                    VALUE B"0".

  01 I        PIC S999 BINARY.

  01 NPOQLENB     PIC S99999 COMP-3 VALUE 670.
  01 NPOQLEN      PIC S99999 COMP-3 VALUE 227.
  01 NPOQNAME.
     05 NPOQNAMEA         PIC X(5) VALUE "NOPAY".
     05 NPOQNAMEWHS       PIC X(4).
     05 NPOQNAMEWHSA REDEFINES NPOQNAMEWHS PIC S9(4).
     05 NPOQNAMEFIL       PIC X(1) VALUE " ".
  01 NPOQLIB         PIC X(10).
  01 NPOQWAIT     PIC S99999 COMP-3 VALUE -1.

  01  TRANSACTION-CONTROL-AREA.
      05  TXN-IND                 PIC X VALUE "0".

  01 CONE   PIC X(1) VALUE "1".
  01 CTWO   PIC X(2) VALUE "2".
  01 CTHREE PIC X(3) VALUE "3".
  01 CFOUR  PIC X(4) VALUE "4".

  01 N1ONE  PIC 9(1) VALUE 1.
  01 NTWO   PIC 9(2) VALUE 2.
  01 NTHREE PIC 9(3) VALUE 3.
  01 NFOUR  PIC 9(4) VALUE 4.

 01  TOO-MANY.
      05 TOO-MANY-1.
         06 TOO-MANY-CID     PIC S9(9) COMP-4.
         06 TOO-MANY-DID     PIC S9(4) COMP-4.
         06 TOO-MANY-WID     PIC S9(4) COMP-4.
      06 TOO-MANY-CLAST      PIC X(16).


     EXEC SQL
       INCLUDE SQLCA
     END-EXEC.


*  SQL ERROR/WARNING Messages  *

     EXEC SQL
       WHENEVER SQLERROR CONTINUE
     END-EXEC.

     EXEC SQL
       WHENEVER SQLWARNING CONTINUE
     END-EXEC.

  01 XX  PIC S9(2) COMP-4.

 LINKAGE SECTION.

 PROCEDURE DIVISION.

 DECLARATIVES.

 HANDLE-ERROR SECTION.

       USE AFTER STANDARD ERROR PROCEDURE ON STOCK-FILE.
       USE AFTER STANDARD ERROR PROCEDURE ON CUST-REL-FILE.

 I-O-ERROR-PARA.

       IF STOCK-FILE-STATUS = "9D"
        SET LOCK-OCCURRED TO TRUE.
       IF CUSTOMER-FILE-STATUS = "9D"
        SET LOCK-OCCURRED TO TRUE.

 END DECLARATIVES.


   MAIN-LINE-ROUTINE.

       PERFORM SET-UP-ROUTINE.

       PERFORM TXN-PROC-STRT THRU
                 TXN-PROC-END
         UNTIL TXN-IND = "1".

       PERFORM CLOSE-ROUTINE THROUGH END-OF-JOB.

       STOP RUN.

*                                                                    ***
*  CMD-KEY ROUTINE PERFORMS ALL DISPLAY FILE I/O AND
*  DETERMINES WHAT TRANSACTION TYPE TO PROCESS.
*                                                                    ***

 TXN-PROC-STRT.

     READ MAINICFF
     INTO TRANSACTION-INPUT INDICATORS ARE CMNF-INDIC-AREA.
     IF MAJ = "81"
         MOVE "1" TO TXN-IND
         GO TO TXN-PROC-END
     END-IF.

D    READ ISO-IFILE.
D    MOVE ISOINUM TO TESTNUM, ISONUM of DBUG-REC, ISONUM OF
D    DBUG-REC2.

       IF TXN-TYPE OF TRANSACTION-INPUT = "N"

              MOVE "0" TO ROLL-BACK-REQUIRED

              PERFORM NEW-ORDER-TRANSACTION
                  THROUGH NEW-ORDER-TRANSACTION-EXIT

             WRITE ICFREC FORMAT IS "SNDDATA"
                   INDICATORS ARE CMNF-INDIC-AREA

          ELSE
          IF TXN-TYPE OF TRANSACTION-INPUT = "P"

              PERFORM PAYMENT-TRANSACTION
                  THROUGH PAYMENT-TRANSACTION-EXIT

             WRITE ICFREC FORMAT IS "SNDDATA"
                   INDICATORS ARE CMNF-INDIC-AREA

          ELSE
          IF TXN-TYPE OF TRANSACTION-INPUT = "O"

              PERFORM ORDER-STATUS-TRANSACTION
                 THROUGH ORDER-STATUS-TRANSACTION-EXIT
```

```
              WRITE ICFREC FORMAT IS "SNDDATA"
                    INDICATORS ARE CMNF-INDIC-AREA

            ELSE

               WRITE ICFREC FORMAT IS "SNDDATA"
                     INDICATORS ARE CMNF-INDIC-AREA
                  MOVE "1" TO TXN-IND

            END-IF.

 TXN-PROC-END.
       EXIT.



 NEW-ORDER-TRANSACTION.

*    ACCEPT CURTIME FROM TIME.
*    ACCEPT DATEINT FROM DATE.
*    MOVE YY TO TPCYEAR.
*    MOVE MMDD TO TPCMD.
     CALL "gettime" USING TIME-OF-DAY.

     MOVE CWID of NEWORD-I to CWID OF CUSTOMER-REC
                           WID OF WAREHOUSE-REC
                          DWID OF DSRCD
                         NOWID OF NEWORDER-REC
                          OWID OF ORDERS-REC.

     MOVE CDID OF NEWORD-I TO
                            CDID OF CUSTOMER-REC
                             DID OF DSRCD
                           NODID OF NEWORDER-REC
                            ODID OF ORDERS-REC.

*    MOVE   CURTIME6 TO OENTTM OF NEWORD-O
*                       OENTTM OF ORDERS-REC.
*    MOVE   TPCDATE  TO OENTDT OF ORDERS-REC.
     MOVE   TIME-OF-DAY TO OENTTOD OF ORDERS-REC.


     MOVE CID OF NEWORD-I TO CID OF CUSTOMER-REC
                             OCID OF ORDERS-REC.

     MOVE NUMBER-OF-ITEMS TO OLINES OF ORDERS-REC.
     MOVE 1 TO OLOCAL OF ORDERS-REC.

     MOVE 0 TO TOTAMT.

 RE-DO-NEWORD.


* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D    MOVE "NEW ORDER PRE-READ  " TO TXTDATA OF DBUG-REC.
D    MOVE 9999 TO OID OF DBUG-REC.
D    PERFORM GET-ISOLATION-DATA-1.
D    WRITE DBUGPRT-REC FORMAT IS "DBUGRCD".


     MOVE CID  OF CUSTOMER-REC TO KVAL OF CSTMR-KEY-DATA(1)
     MOVE CDID OF CUSTOMER-REC TO KVAL OF CSTMR-KEY-DATA(2)
     MOVE CWID OF CUSTOMER-REC TO KVAL OF CSTMR-KEY-DATA(3)

      CALL "qdbrunha" USING BY VALUE
           CSTMR-HASH-PTR FETCHUPDATE CUST-KEYS CSTMR-KEY-PTR
           CSTMR-DEF-PTR RET-CODE-PTR

     IF RET-CODE > 0
           IF RET-CODE = 100
                MOVE "2" TO ROLL-BACK-REQUIRED
           ELSE
               IF RET-CODE = 812
                   SET LOCK-OFF TO TRUE
                   ROLLBACK
                   GO TO RE-DO-NEWORD
               END-IF
            END-IF

     ELSE


     READ DISTRICT-FILE
     INVALID KEY
          MOVE "2" TO ROLL-BACK-REQUIRED

     NOT INVALID KEY
     MOVE DNXTOR OF DISTRICT-REC   TO OID OF ORDERS-REC
                                      NOOID OF NEWORDER-REC

     PERFORM VARYING I FROM 1 BY 1 UNTIL I > NUMBER-OF-ITEMS
           MOVE INPUT-LINE(I) TO INPUT-ROW
           MOVE OLIID OF INPUT-ROW TO IID OF ITEM-REC


     MOVE OLIID OF INPUT-ROW TO KVAL OF ITEM-KEY-DATA(1)

      CALL "qdbrunha" USING BY VALUE
           ITEM-HASH-PTR FETCHR ITEM-KEYS ITEM-KEY-PTR
           ITEM-DEF-PTR RET-CODE-PTR

           IF RET-CODE > 0
               MOVE "1" TO ROLL-BACK-REQUIRED

           ELSE

               MOVE ITEM-INFO TO  OUTPUT-ITEM-INFO(I)

               MOVE ZERO TO CONTR-1

               INSPECT IDATA OF ITEM-REC
               TALLYING CONTR-1 FOR ALL "ORIGINAL"

               MOVE STOCK-KEY-INPUT TO STOCK-IN

     MOVE STWID1             TO KVAL OF STOCK-KEY-DATA(1)
     MOVE STIID1             TO KVAL OF STOCK-KEY-DATA(2)

     CALL "qdbrunha" USING BY VALUE
           STOCK-HASH-PTR FETCHUPDATE STOCK-KEYS STOCK-KEY-PTR
           STOCK-DEF-PTR RET-CODE-PTR

          IF RET-CODE > 0
           IF RET-CODE = 100
                MOVE "2" TO ROLL-BACK-REQUIRED
           ELSE
               IF RET-CODE = 812
                   SET LOCK-OFF TO TRUE
                   ROLLBACK
                   GO TO RE-DO-NEWORD
                END-IF
              END-IF
             ELSE


* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D              IF TESTNUM = 7 THEN
D                IF I = 1 THEN
D                  CALL "delayme"
D                END-IF
D              END-IF


                SUBTRACT OLQTY OF INPUT-ROW
                                  FROM STQTY OF STOCK-REC
                IF STQTY OF STOCK-REC < 10 THEN
                     ADD 91 TO STQTY OF STOCK-REC
                END-IF


                IF (OLSPWH OF INPUT-ROW NOT EQUAL CWID OF
                                                    NEWORD-I)
                     MOVE 0 TO OLOCAL   OF ORDERS-REC
                     ADD  1 TO STREMORD OF STOCK-REC
                END-IF

                ADD 1              TO STORDRS OF STOCK-REC
                ADD OLQTY OF INPUT-ROW TO STYTD OF STOCK-REC

           CALL "qdbrunha" USING BY VALUE
                STOCK-HASH-PTR UPDATER STOCK-KEYS STOCK-KEY-PTR
                STOCK-DEF-PTR RET-CODE-PTR
                   IF CONTR-1 > ZERO THEN
                      MOVE ZERO TO CONTR-1
                      INSPECT STDATA OF STOCK-REC
                      TALLYING CONTR-1 FOR ALL "ORIGINAL"
                   END-IF

                   IF CONTR-1 > ZERO
                   THEN MOVE "B" TO OUTPUT-BORG(I)
                   ELSE MOVE "G" TO OUTPUT-BORG(I)
                   END-IF

                MULTIPLY OLQTY OF INPUT-ROW BY IPRICE
                     GIVING OUTPUT-OLAMNT(I)

                ADD OUTPUT-OLAMNT(I) TO TOTAMT

                MOVE NORCD            TO ORDER-KEY
                MOVE I                TO OLNBR OF OLRCD
                                         OF ORDERLINE-REC
                MOVE INPUT-ROW    TO OL-INPUT
                MOVE OUTPUT-OLAMNT(I) TO OLAMNT OF OLRCD
                                         OF ORDERLINE-REC
                MOVE DIST-INFO(CDID OF NEWORD-I)
                                      TO OLDSTI OF OLRCD
                                         OF ORDERLINE-REC

                MOVE DATETIME-CHARINIT TO OLTIME-OF-DAY
                WRITE ORDERLINE-REC

                MOVE STQTY OF STOCK-REC TO OUTPUT-STQTY(I)
              END-IF
              END-IF
            END-PERFORM
           END-READ
          END-IF.

D    MOVE OID OF ORDERS-REC TO OID OF DBUG-REC.

     ADD 1 TO DNXTOR OF DISTRICT-REC.

     REWRITE DISTRICT-REC.

     WRITE ORDERS-REC.

     WRITE NEWORDER-REC.

     READ WAREHOUSE-FILE.

     IF ROLL-BACK-REQUIRED NOT EQUAL "0"
     THEN
* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D         MOVE "NEW ORDER PRE-RBACK " TO TXTDATA OF DBUG-REC
D         PERFORM GET-ISOLATION-DATA-1
D         WRITE DBUGPRT-REC FORMAT IS "DBUGRCD"
D         CALL "delayme"

         ROLLBACK
* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D         MOVE "NEW ORDER POST-RBACK" TO TXTDATA OF DBUG-REC
D         PERFORM GET-ISOLATION-DATA-2
D         WRITE DBUGPRT-REC FORMAT IS "DBUGRCD"
D         CLOSE DBUG-FILE
D         OPEN OUTPUT DBUG-FILE


          MOVE CLAST OF CUSTOMER-REC
                    TO CLAST OF NEWORD-O
          MOVE CCREDT OF CUSTOMER-REC
                    TO CCREDT OF NEWORD-O
          MOVE OID OF ORDERS-REC
                    TO OID OF NEWORD-O
        IF ROLL-BACK-REQUIRED = "1"
          MOVE "R" TO NEWORD-RESULT of NEWORD-O
        ELSE IF ROLL-BACK-REQUIRED = "2"
          MOVE "I" TO NEWORD-RESULT of NEWORD-O
        END-IF
        END-IF
          MOVE "0" TO ROLL-BACK-REQUIRED
          GO TO NEW-ORDER-TRANSACTION-EXIT
      ELSE
* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D         MOVE "NEW ORDER PRE-COMMIT" TO TXTDATA OF DBUG-REC
D         PERFORM GET-ISOLATION-DATA-1
D         WRITE DBUGPRT-REC FORMAT IS "DBUGRCD"
D         CALL "delayme"



          COMMIT.

* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D         MOVE "NEW ORDER POST-COMIT" TO TXTDATA OF DBUG-REC.
```

```
D         PERFORM GET-ISOLATION-DATA-2.
D         WRITE DBUGPRT-REC FORMAT IS "DBUGRCD".
D         CLOSE DBUG-FILE.
D         OPEN OUTPUT DBUG-FILE.

     MOVE "G" TO NEWORD-RESULT of NEWORD-O
     MOVE CLAST OF CUSTOMER-REC
           TO CLAST OF NEWORD-O
     MOVE CCREDT OF CUSTOMER-REC
           TO CCREDT OF NEWORD-O
     MOVE CDCT of CUSTOMER-REC TO CDCT OF NEWORD-O
     MOVE WTAX of WAREHOUSE-REC TO WTAX OF NEWORD-O
     MOVE DTAX of DISTRICT-REC TO DTAX OF NEWORD-O
     MOVE OID OF ORDERS-REC TO
               OID OF NEWORD-O.


 NEW-ORDER-TRANSACTION-EXIT.

     EXIT.


D GET-ISOLATION-DATA-1.
D      ACCEPT CURTIME FROM TIME
D      ACCEPT DATEINT FROM DATE
D      MOVE YY TO TPCYEAR
D      MOVE MMDD TO TPCMD
D      MOVE CDID OF NEWORD-I TO DID OF DBUG-REC
D      MOVE CID OF NEWORD-I TO CID OF DBUG-REC
D      MOVE cwid of NEWORD-I TO WID OF DBUG-REC
D      MOVE TPCDATE TO DATE6 OF DBUG-REC
D      MOVE CURTIME6 TO TIME6 OF DBUG-REC
D      MOVE TESTNUM TO ISONUM OF DBUG-REC
D      PERFORM VARYING J FROM 1 BY 1 UNTIL J = 16
D        MOVE OLSPWH OF INPUT-LINE(J) TO DEBUG-OLSPWH(J)
D        MOVE OLIID OF INPUT-LINE(J) TO DEBUG-OLIID(J)
D        MOVE OLQTY OF INPUT-LINE(J) TO DEBUG-OLQTY(J)
D        MOVE 0 TO DEBUG-IPRICE(J)
D      END-PERFORM.


D GET-ISOLATION-DATA-2.
D      MOVE OID OF ORDERS-REC TO OID OF DBUG-REC.
D      ACCEPT CURTIME FROM TIME
D      ACCEPT DATEINT FROM DATE
D      MOVE YY TO TPCYEAR
D      MOVE MMDD TO TPCMD
D      MOVE CDID OF NEWORD-I TO DID OF DBUG-REC
D      MOVE CID OF NEWORD-I TO CID OF DBUG-REC
D      MOVE cwid of NEWORD-I TO WID OF DBUG-REC
D      MOVE TPCDATE TO DATE6  OF DBUG-REC
D      MOVE CURTIME6 TO TIME6  OF DBUG-REC
D      MOVE TESTNUM TO ISONUM OF DBUG-REC
D      PERFORM VARYING J FROM 1 BY 1 UNTIL J = 16
D        MOVE OLSPWH OF INPUT-LINE(J) TO DEBUG-OLSPWH(J)
D        MOVE OLIID OF INPUT-LINE(J) TO DEBUG-OLIID(J)
D        MOVE OLQTY OF INPUT-LINE(J) TO DEBUG-OLQTY(J)
D        IF OLIID OF INPUT-LINE(J) <= 100000
D        MOVE OUTPUT-IPRICE(J) TO DEBUG-IPRICE(J)
D        ELSE
D        MOVE 0 TO DEBUG-IPRICE(J)
D        END-IF
D      END-PERFORM.


D GET-ISOLATION-DATA-3.
D      ACCEPT CURTIME FROM TIME
D      ACCEPT DATEINT FROM DATE
D      MOVE YY TO TPCYEAR
D      MOVE MMDD TO TPCMD
D      MOVE DID OF ORDSTS-I TO DID OF DBUG-REC
D      MOVE CID OF ORDSTS-I TO CID OF DBUG-REC
D      MOVE WID of ORDSTS-I TO WID OF DBUG-REC
D      MOVE TPCDATE TO DATE6 OF DBUG-REC
D      MOVE CURTIME6 TO TIME6 OF DBUG-REC
D      MOVE TESTNUM TO ISONUM OF DBUG-REC
D      PERFORM VARYING J FROM 1 BY 1 UNTIL J = 16
D        MOVE 0 TO DEBUG-OLSPWH(J)
D        MOVE 0 TO DEBUG-OLIID(J)
D        MOVE 0 TO DEBUG-OLQTY(J)
D        MOVE 0 TO DEBUG-IPRICE(J)
D      END-PERFORM.


D GET-ISOLATION-DATA-4.
D      ACCEPT CURTIME FROM TIME
D      ACCEPT DATEINT FROM DATE
D      MOVE YY TO TPCYEAR
D      MOVE MMDD TO TPCMD
D      MOVE DID OF ORDSTS-I TO DID OF DBUG-REC
D      MOVE CID OF ORDSTS-I TO CID OF DBUG-REC
D      MOVE WID of ORDSTS-I TO WID OF DBUG-REC
D      MOVE TPCDATE TO DATE6 OF DBUG-REC
D      MOVE CURTIME6 TO TIME6 OF DBUG-REC
D      MOVE TESTNUM TO ISONUM OF DBUG-REC
D      PERFORM VARYING J FROM 1 BY 1 UNTIL J = 16
D        MOVE DO-OLSPWH(J) TO DEBUG-OLSPWH(J)
D        MOVE DO-OLIID(J) TO DEBUG-OLIID(J)
D        MOVE DO-OLQTY(J) TO DEBUG-OLQTY(J)
D        MOVE 0 TO DEBUG-IPRICE(J)
D      END-PERFORM.



 PAYMENT-TRANSACTION.

     MOVE HISTORY-DATA OF PAYMENT-I TO HISTORY-DATA OF HIST-INFO.
     MOVE AMOUNT OF PAYMENT-I TO AMOUNT OF HIST-INFO.

*    ACCEPT DATEINT FROM DATE.
*    MOVE YY TO TPCYEAR.
*    MOVE MMDD TO TPCMD.
*    ACCEPT CURTIME FROM TIME.
     CALL "gettime" USING TIME-OF-DAY.

*    MOVE  DATETIME TO DATE-TIME OF HIST-INFO, HDT, ODT.
     MOVE  TIME-OF-DAY TO DATE-TIME OF HIST-INFO, HDT, ODT.

* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D    MOVE "PAY PRE-READ  " TO TXTDATA OF DBUG-REC2.
D    MOVE CDID OF CUSTOMER-REC TO CDID OF DBUG-REC2.
D    MOVE CWID OF CUSTOMER-REC TO CWID OF DBUG-REC2.
D    MOVE CID OF PAYMENT-I TO CID OF DBUG-REC2.
D    MOVE CDID OF CUSTOMER-REC TO DID OF DBUG-REC2.
D    MOVE AMOUNT OF PAYMENT-I TO
D           PAYMNT OF DBUG-REC2.
D    MOVE WID OF PAYMENT-I TO WID OF DBUG-REC2.
D    MOVE 0 TO CBAL OF DBUG-REC2.
D    MOVE CURTIME6 TO TIME6 OF DBUG-REC2.
D    MOVE TPCDATE TO DATE6 OF DBUG-REC2.
D    WRITE DBUGPRT-REC2 FORMAT IS "DBUGRCD2".


     IF PAYMENT-TYPE EQUAL "C"
     MOVE PAY-CUST-KEY OF PAYMENT-I TO CUST-KEY
          OF CUSTOMER-REC
        PERFORM CUSTOMER-BY-NUMBER
     ELSE
     MOVE CWID OF PAYMENT-I TO CWID OF CSRCD OF CUST-REL-REC
     MOVE CDID OF PAYMENT-I TO CDID OF CSRCD OF CUST-REL-REC
     MOVE CLAST OF PAYMENT-I TO CLAST OF CUST-REL-REC


           PERFORM CUSTOMER-BY-NAME THRU CUSTOMER-BY-NAME-EXIT.

D    MOVE CBAL OF CSRCD OF CUSTOMER-REC TO CBAL OF DBUG-REC2.
     SUBTRACT AMOUNT OF PAYMENT-I
              FROM CBAL OF CSRCD OF CUSTOMER-REC.

     ADD AMOUNT OF PAYMENT-I
         TO CYTD OF CUSTOMER-REC.

     ADD 1 TO CPAYCNT OF CUSTOMER-REC.
     MOVE CUST-INFO TO PAYMENT-CUST-INFO.
     MOVE CBAL OF CSRCD OF CUSTOMER-REC TO CBAL OF PAYMNT-O.
     MOVE CCRDLM OF CSRCD OF CUSTOMER-REC
          TO CCRDLM OF PAYMNT-O.
     MULTIPLY CDCT OF CSRCD OF CUSTOMER-REC BY 100 GIVING
          CDCT OF PAYMNT-O.

     MOVE SPACES                  TO MISC-CDATA.
     IF CCREDT OF CSRCD OF CUSTOMER-REC = "BC" THEN
       MOVE CDATA OF CSRCD OF CUSTOMER-REC (1:468)
            TO TEMP-DATA2
       MOVE TEMP-DATA2
            TO CDATA OF CSRCD OF CUSTOMER-REC (32:468)

       MOVE DID OF HISTORY-DATA OF HIST-INFO
                                 TO DID OF CHAR-HIST-DATA
       MOVE WID OF HISTORY-DATA OF HIST-INFO
                                TO WID OF CHAR-HIST-DATA
       MOVE CID OF HISTORY-DATA OF HIST-INFO
                                TO CID OF CHAR-HIST-DATA
       MOVE CDID OF HISTORY-DATA OF HIST-INFO
                                 TO CDID OF CHAR-HIST-DATA
       MOVE CWID OF HISTORY-DATA OF HIST-INFO
                                 TO CWID OF CHAR-HIST-DATA
       MOVE AMOUNT OF HIST-INFO
                                    TO AMOUNT OF CHAR-HIST-DATA
       MOVE CHAR-HIST-DATA TO CDATA OF CUSTOMER-REC(1:31)
       MOVE CDATA OF CUSTOMER-FILE TO MISC-CDATA
     END-IF.
     IF PAYMENT-TYPE EQUAL "C"
          CALL "qdbrunha" USING BY VALUE
                CSTMR-HASH-PTR UPDATER CUST-KEYS CSTMR-KEY-PTR
                CSTMR-DEF-PTR RET-CODE-PTR
     ELSE
     MOVE CUSTOMER-REC TO CUST-REL-REC
            REWRITE CUST-REL-REC

     END-IF.
     MOVE  PAY-DIST-KEY OF PAYMENT-I TO DIST-KEY.

     READ DISTRICT-FILE INVALID KEY
          MOVE "77" TO OUTPUT-FMT-NUM-3
          ROLLBACK
          GO TO PAYMENT-TRANSACTION-EXIT.

     ADD AMOUNT OF PAYMENT-I
         TO DYTD OF DISTRICT-REC.

     REWRITE DISTRICT-REC.

     MOVE WID OF PAYMENT-I TO WID OF WAREHOUSE-REC.


     READ WAREHOUSE-FILE.

     ADD AMOUNT OF PAYMENT-I
         TO WYTD OF WAREHOUSE-REC.

     REWRITE WAREHOUSE-REC.

     MOVE CID OF CUSTOMER-REC TO
              CID OF HISTORY-REC.
     MOVE WNAME OF WAREHOUSE-REC TO
              WNAME OF HDATA OF HIST-INFO.
     MOVE DNAME OF DISTRICT-REC TO
             DNAME OF HDATA OF HIST-INFO.

     WRITE HISTORY-REC.

* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D    MOVE "PAY PRE-COMMIT" TO TXTDATA OF DBUG-REC2.
D    MOVE CDID OF CUSTOMER-REC TO CDID OF DBUG-REC2.
D    MOVE CWID OF CUSTOMER-REC TO CWID OF DBUG-REC2.
D    MOVE CID OF PAYMENT-I TO CID OF DBUG-REC2.
D    MOVE CDID OF CUSTOMER-REC TO DID OF DBUG-REC2.
D    MOVE WID OF PAYMENT-I TO WID OF DBUG-REC2.
*    MOVE AMOUNT OF PAYMENT-I TO
*           PAYMNT OF DBUG-REC2.
D    ACCEPT CURTIME FROM TIME.
D    MOVE CURTIME6 TO TIME6 OF DBUG-REC2.
D    MOVE TPCDATE TO DATE6 OF DBUG-REC2.
D    WRITE DBUGPRT-REC2 FORMAT IS "DBUGRCD2".
D                  CALL "delayme".


     COMMIT.

* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D    MOVE "PAY POST-COMMIT" TO TXTDATA OF DBUG-REC2.
D    MOVE CDID OF CUSTOMER-REC TO CDID OF DBUG-REC2.
D    MOVE CWID OF CUSTOMER-REC TO CWID OF DBUG-REC2.
D    MOVE CID OF PAYMENT-I TO CID OF DBUG-REC2.
D    MOVE CDID OF CUSTOMER-REC TO DID OF DBUG-REC2.
*    MOVE AMOUNT OF PAYMENT-I TO
*           PAYMNT OF DBUG-REC2.
D    MOVE WID OF PAYMENT-I TO WID OF DBUG-REC2.
D    MOVE CBAL OF CSRCD OF CUSTOMER-REC TO CBAL OF DBUG-REC2.
D    ACCEPT CURTIME FROM TIME.
D    MOVE CURTIME6 TO TIME6 OF DBUG-REC2.
D    MOVE TPCDATE TO DATE6 OF DBUG-REC2.
D    WRITE DBUGPRT-REC2 FORMAT IS "DBUGRCD2".
D    CLOSE DBUG-FILE2.
D    OPEN OUTPUT DBUG-FILE2.


     MOVE DIST-ADDRESS TO DST-DATA.
     MOVE WH-ADDRESS TO WH-DATA.
```

```
    MOVE CID OF CUSTOMER-REC TO CID OF PAYMNT-O.

    IF CCREDT OF CUSTOMER-REC = "BC"
    THEN MOVE "17" TO OUTPUT-FMT-NUM-3
    ELSE MOVE "19" TO OUTPUT-FMT-NUM-3
    END-IF.

END-WAREHOUSE-UPDATE.
PAYMENT-TRANSACTION-EXIT.

CUSTOMER-BY-NUMBER.
        MOVE CID  OF CUSTOMER-REC TO KVAL OF CSTMR-KEY-DATA(1)
        MOVE CDID OF CUSTOMER-REC TO KVAL OF CSTMR-KEY-DATA(2)
        MOVE CWID OF CUSTOMER-REC TO KVAL OF CSTMR-KEY-DATA(3)

        CALL "qdbrunha" USING BY VALUE
             CSTMR-HASH-PTR FETCHUPDATE CUST-KEYS CSTMR-KEY-PTR
             CSTMR-DEF-PTR RET-CODE-PTR

        IF RET-CODE > 0
          IF RET-CODE = 100
            MOVE "77" TO OUTPUT-FMT-NUM-3
            ROLLBACK
            GO TO PAYMENT-TRANSACTION-EXIT
          ELSE
              IF RET-CODE = 812
                  SET LOCK-OFF TO TRUE
                  GO TO CUSTOMER-BY-NUMBER
              END-IF
           END-IF
         END-IF.

CUSTOMER-BY-NAME.

     MOVE CWID OF CUST-REL-REC TO CWIDR.
     MOVE CDID OF CUST-REL-REC TO CDIDR.
     MOVE CLAST OF CUST-REL-REC TO CLASTR.

     EXEC SQL DECLARE C1 CURSOR FOR
     SELECT RRN(CSTMRPF)
     FROM CSTMRPF
     WHERE CLAST = :CLASTR AND CDID = :CDIDR AND
     CWID = :CWIDR
     END-EXEC.

     EXEC SQL OPEN C1
     END-EXEC.

     EXEC SQL FETCH C1 FOR 100 ROWS INTO :HOST-ARRAY-ELEMENT
     END-EXEC.
     MOVE SQLERRD OF SQLCA(3) TO XX.

     EXEC SQL CLOSE C1
     END-EXEC.

     IF XX = 0
         IF TXN-TYPE = 'P'
         MOVE "77" TO OUTPUT-FMT-NUM-3
         ROLLBACK
         GO TO payment-transaction-EXIT
         END-IF
         IF TXN-TYPE = 'O'
         MOVE "62" TO ORDSTS-FMT-NUM
         ROLLBACK
         GO TO ORDER-STATUS-TRANSACTION-EXIT
         END-IF
     END-IF

     IF XX = 100
         CALL "TOOMANY" USING CWIDR, CDIDR, CLASTR, CUSTREL
     ELSE
         COMPUTE CUST-INDEX = (XX + 1) / 2.
         MOVE HOST-ARRAY-ELEMENT(CUST-INDEX) TO CUSTREL.

CUSTOMER-BY-NAME-EXIT.

    READ CUST-REL-FILE.
              IF LOCK-OCCURRED
                  SET LOCK-OFF TO TRUE
                  ROLLBACK
                  GO TO CUSTOMER-BY-NAME
              END-IF
    MOVE CUST-REL-REC TO CUSTOMER-REC.




 ORDER-STATUS-TRANSACTION.

* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D    MOVE "ORDER STATUS PRE-READ  " TO TXTDATA OF DBUG-REC.
D    MOVE 0 TO OID OF DBUG-REC
D    PERFORM GET-ISOLATION-DATA-3.
D    WRITE DBUGPRT-REC FORMAT IS "DBUGRCD".

     MOVE WID OF ORDSTS-I TO OWID OF ORDERS-VIEW-REC.
     MOVE DID OF ORDSTS-I TO ODID OF ORDERS-VIEW-REC.
     MOVE ORD-CUST-KEY OF ORDSTS-I TO CUST-KEY OF CUSTOMER-REC.
 ORDER-STATUS-READS.
     IF ordsts-type = "C" THEN
         MOVE CID  OF CUSTOMER-REC TO KVAL OF CSTMR-KEY-DATA(1)
         MOVE CDID OF CUSTOMER-REC TO KVAL OF CSTMR-KEY-DATA(2)
         MOVE CWID OF CUSTOMER-REC TO KVAL OF CSTMR-KEY-DATA(3)

         CALL "qdbrunha" USING BY VALUE
              CSTMR-HASH-PTR FETCHUPDATE CUST-KEYS CSTMR-KEY-PTR
              CSTMR-DEF-PTR RET-CODE-PTR
        IF RET-CODE > 0
          IF RET-CODE = 100
                MOVE "62" TO ORDSTS-FMT-NUM
                GO TO ORDER-STATUS-TRANSACTION-EXIT
           ELSE
               IF RET-CODE = 812
                   SET LOCK-OFF TO TRUE
                   GO TO ORDER-STATUS-READS
               END-IF
            END-IF
         END-IF
     ELSE
        MOVE CLAST OF ORDSTS-I TO CLAST OF CUST-REL-REC
        MOVE CLAST OF CUST-REL-REC TO CLAST OF PAYMENT-I
        MOVE WID OF ORDSTS-I TO CWID OF CUST-REL-REC
        MOVE DID OF ORDSTS-I TO CDID OF CUST-REL-REC
        PERFORM CUSTOMER-BY-NAME THROUGH
           CUSTOMER-BY-NAME-EXIT
     END-IF.
     MOVE CID OF CUSTOMER-REC TO OCID OF ORDERS-VIEW-REC
                                 OCID OF ORDSTS-O.
     MOVE 999999 TO OID OF ORDERS-VIEW-REC.

     START ORDERS-VIEW KEY NOT < EXTERNALLY-DESCRIBED-KEY
         INVALID KEY
                 MOVE "62" TO ORDSTS-FMT-NUM
                 GO TO ORDER-STATUS-TRANSACTION-EXIT.
     READ ORDERS-VIEW PRIOR
D                IF TESTNUM = 9 THEN
D                   CALL "delayme"
D                END-IF

     MOVE OWID OF ORDERS-VIEW-REC TO
            OLWID OF ORDERLINE-VIEW-REC.
     MOVE ODID OF ORDERS-VIEW-REC TO
            OLDID OF ORDERLINE-VIEW-REC.
     MOVE OID OF ORDERS-VIEW-REC TO
            OLOID OF ORDERLINE-VIEW-REC OID OF ORDSTS-O.
     MOVE 1 TO OLNBR OF ORDERLINE-VIEW-REC.
     MOVE OLINES OF ORDERS-VIEW-REC TO
            OLINENBR OF ORDSTS-O.
     MOVE OCARID OF ORDERS-VIEW-REC TO
            OCARID OF ORDSTS-O.
*    MOVE OENTDT OF ORDERS-VIEW-REC TO
*           OENTDT OF ORDSTS-O.
     MOVE OENTTOD OF ORDERS-VIEW-REC TO
            OENTTOD OF ORDSTS-O.
*    MOVE OENTTM OF ORDERS-VIEW-REC TO
*           OENTTM OF ORDSTS-O.

     READ ORDERLINE-VIEW
     MOVE OL-INFO TO OLINEINFO(1)

     PERFORM varying I
             from 2 by 1 until I > OLINES OF ORDERS-VIEW-REC

         READ ORDERLINE-VIEW NEXT
         MOVE OL-INFO TO OLINEINFO(I)

     END-PERFORM.

       MOVE I TO ORDSTS-FMT-NUM.
D      MOVE OID OF ORRCD OF ORDERS-VIEW-REC TO OID OF DBUG-REC.
     COMMIT.

* THE FOLLOWING CODE WILL ONLY BE INCLUDED WHEN THE
* FLAG IS SET AND IS FOR ISOLATION TEST USE
*
D    MOVE "ORDER STATUS POST-COMMIT " TO TXTDATA OF DBUG-REC.
*    MOVE 0 TO OID OF DBUG-REC
D    PERFORM GET-ISOLATION-DATA-4.
D    WRITE DBUGPRT-REC FORMAT IS "DBUGRCD".
D    CLOSE DBUG-FILE.
D    OPEN OUTPUT DBUG-FILE.


     MOVE CBAL OF CSRCD OF CUSTOMER-REC TO CBAL OF ORDSTS-O.
     MOVE CFIRST OF CUSTOMER-REC TO CFIRST OF ORDSTS-O.
     MOVE CINIT OF CUSTOMER-REC TO CINIT OF ORDSTS-O.
     MOVE CLAST OF CUSTOMER-REC TO CLAST OF ORDSTS-O.

 ORDER-STATUS-TRANSACTION-EXIT.
       EXIT.




 SET-UP-ROUTINE.

*     ACCEPT DATEINT FROM DATE.
*     MOVE YY TO TPCYEAR.
*     MOVE MMDD TO TPCMD.
*     IF YY < 70 THEN
*        MOVE 20 TO TPCCEN.
     CALL "gettime" USING TIME-OF-DAY.


*OPEN FILES
     OPEN  I-O   DISTRICT-FILE
                 WAREHOUSE-FILE
                 STOCK-FILE
                 CUSTOMER-FILE
                 CUST-REL-FILE
                 MAINICFF.
     OPEN INPUT  ITEM-FILE
                 ORDERLINE-VIEW
                 ORDERS-VIEW.
     OPEN EXTEND ORDERLINE-FILE.
     OPEN EXTEND ORDERS-FILE.
     OPEN EXTEND NEWORDER-FILE.
D    OPEN INPUT  ISO-IFILE.
D    OPEN OUTPUT DBUG-FILE.
D    OPEN OUTPUT DBUG-FILE2.
     OPEN EXTEND HISTORY-FILE.

     MOVE "ITEMLF"           TO ITEM-HASH-NAME.
     MOVE "IID"              TO KNAM OF ITEM-KEY-DATA(1).

     SET ITEM-HASH-PTR TO ADDRESS OF ITEM-HASH-NAME.
     SET RET-CODE-PTR TO ADDRESS OF RET-CODE.
     SET ITEM-DEF-PTR TO ADDRESS OF ITEM-DEF.
     SET ITEM-KEY-PTR TO ADDRESS OF ITEM-KEY.

     MOVE "STOCK"            TO STOCK-HASH-NAME.
     MOVE "STIID"            TO KNAM OF STOCK-KEY-DATA(2).
     MOVE "STWID"            TO KNAM OF STOCK-KEY-DATA(1).

     SET STOCK-HASH-PTR TO ADDRESS OF STOCK-HASH-NAME.
     SET RET-CODE-PTR  TO ADDRESS OF RET-CODE.
     SET STOCK-DEF-PTR TO ADDRESS OF STOCK-DEF.
     SET STOCK-KEY-PTR TO ADDRESS OF STOCK-KEY.

     MOVE "CSTMR"            TO CSTMR-HASH-NAME.
     MOVE "CID"              TO KNAM OF STOCK-KEY-DATA(1).
     MOVE "CDID"             TO KNAM OF STOCK-KEY-DATA(2).
     MOVE "CWID"             TO KNAM OF STOCK-KEY-DATA(3).

     SET CSTMR-HASH-PTR TO ADDRESS OF CSTMR-HASH-NAME.
     SET RET-CODE-PTR   TO ADDRESS OF RET-CODE.
     SET CSTMR-DEF-PTR  TO ADDRESS OF CUSTOMER-REC.
     SET CSTMR-KEY-PTR  TO ADDRESS OF CSTMR-KEY.

     MOVE SPACES TO HDATA OF HIST-INFO
                    OCARID OF ORDERS-REC.
*    MOVE ZEROES TO OLDLVT OF ORDERLINE-REC.
*                   OLDLVT OF ORDERLINE-REC.
     MOVE ZEROES TO OLDLVTOD OF ORDERLINE-REC.

 CLOSE-ROUTINE.
*CLOSE ALL FILES
     CLOSE       DISTRICT-FILE
                 STOCK-FILE
                 CUSTOMER-FILE
                 ITEM-FILE
```

```
                WAREHOUSE-FILE
                CUST-REL-FILE
                ORDERLINE-VIEW
                ORDERS-VIEW
                ORDERLINE-FILE
                ORDERS-FILE
                NEWORDER-FILE
D               DBUG-FILE
D               DBUG-FILE2
                HISTORY-FILE
                MAINICFF.

 END-OF-JOB.
     STOP RUN.
```

TOOMANY: Handles More Than 100 Customer Last Names

```
 PROCESS NOTRUNC NORANGE.
 IDENTIFICATION DIVISION.

*
*  TOO MANY Program
*
*     Files Accessed:   - CSTMR     (Read)
*     Views Used:       - CSTMRLFNAM (C1 - SQL)
*
    PROGRAM-ID. TOOMANY.
    AUTHOR. DEPT-53G.
    INSTALLATION. IBM-ROCHESTER.
    DATE-WRITTEN. 09-03-95.
    DATE-COMPILED. 12-14-92.
 ENVIRONMENT DIVISION.
 CONFIGURATION SECTION.
    SOURCE-COMPUTER. IBM-AS400.
** SOURCE-COMPUTER. IBM-AS400. WITH DEBUGGING MODE.
    OBJECT-COMPUTER. IBM-AS400.

 DATA DIVISION.

****************************
 WORKING-STORAGE SECTION.
****************************

 01 HOST-VARIABLES.
     05 VWID                      PIC S9(4) COMP-4.
     05 VDID                      PIC S9(2) COMP-4.
     05 VCID                      PIC S9(6) COMP-4.
     05 VCLAST                    PIC X(16).
     05 VCFIRST                   PIC X(16).
     05 VCOUNT                      PIC S9(4) COMP-4.
     05 RRN    PIC S9(9) COMP-4.
 01 I                             PIC S9(4) COMP-4.

*****************
*  SQL INCLUDE  *
*****************
      EXEC SQL
        INCLUDE SQLCA
      END-EXEC.

*********************************
*  SQL ERROR/WARNING Messages  *
*********************************
      EXEC SQL
        WHENEVER SQLERROR CONTINUE
      END-EXEC.

      EXEC SQL
        WHENEVER SQLWARNING CONTINUE
      END-EXEC.

*******************
 LINKAGE SECTION.
*******************
  01 CWIDR    PIC S9(4) COMP-4.
  01 CDIDR    PIC S9(2) COMP-4.
  01 CLASTR   PIC X(16).
  01 CUSTREL  PIC 9(8) COMP-4.

 PROCEDURE DIVISION USING CWIDR, CDIDR, CLASTR, CUSTREL.

********************
 START-OF-PROGRAM.
********************

      MOVE CWIDR TO VWID.
      MOVE CDIDR TO VDID.
      MOVE CLASTR TO VCLAST.

      EXEC SQL
      SELECT COUNT(*)
      INTO :.VCOUNT
      FROM CSTMRPF
      WHERE (CWID = :.VWID
            AND CDID = :VDID AND CLAST = :.VCLAST)
      END-EXEC.

      COMPUTE VCOUNT = 10 * VCOUNT + 1.
      DIVIDE VCOUNT BY 20 GIVING VCOUNT ROUNDED.

      EXEC SQL DECLARE C1 CURSOR FOR
      SELECT RRN(CSTMRPF)
      FROM CSTMRPF
      WHERE CLAST = :.VCLAST AND CDID = :.VDID AND
      CWID = :.VWID
      END-EXEC.

      EXEC SQL OPEN C1
      END-EXEC.

      PERFORM VARYING I FROM 1 BY 1 UNTIL I > VCOUNT

      EXEC SQL FETCH C1 INTO :.RRN
      END-EXEC

      END-PERFORM.

      EXEC SQL CLOSE C1
      END-EXEC.

      MOVE RRN TO CUSTREL.


     EXIT.
```

ATPCCMTRS: Stock Level Start Program

```
            PGM

            DCL        VAR(&WRHS) TYPE(*CHAR) LEN(4)

            DCL        VAR(&QLIB) TYPE(*CHAR) LEN(10)
            DCL        VAR(&DATALIB) TYPE(*CHAR) LEN(10) +
                         VALUE(TPCCD697)
            DCL        VAR(&APPLIB) TYPE(*CHAR) LEN(10) +
                         VALUE(TPCCNCSRVR)


            MONMSG     MSGID(CPF0000)
            CHGJOB     RUNPTY(20) LOG(4 0 *SECLVL) LOGCLPGM(*YES)

  /*        RTVDTAARA  DTAARA(TPCCINFO/DATALIB) RTNVAR(&DATALIB) */
            ADDLIBLE   LIB(&APPLIB) /* TPC-C application programs +
                         and display file */
            MONMSG     MSGID(CPF0000)
            ADDLIBLE   LIB(&DATALIB) /* TPC-C data base library */
            MONMSG     MSGID(CPF0000)

            OVRDBF     FILE(ORDLINPF) TOFILE(ORDLINPF) +
                         NBRRCDS(16) LVLCHK(*NO) +
                         SEQONLY(*YES 16)

            OVRDBF     FILE(ORDLINLF) TOFILE(ORDLINLF) +
                         NBRRCDS(16) LVLCHK(*NO) +
                         SEQONLY(*YES 16)

            OVRDBF     FILE(ORDLIN) TOFILE(ORDLIN) +
                         NBRRCDS(16) LVLCHK(*NO) +
                         SEQONLY(*YES 16)

            OVRDBF     FILE(DSTRCT) TOFILE(DSTRCT) LVLCHK(*NO) +
                         SHARE(*YES)

            OVRDBF     FILE(STOCKPF) TOFILE(STOCKPF) WAITRCD(2000) +
                         LVLCHK(*NO)

            OVRDBF     FILE(STOCK) TOFILE(STOCK) LVLCHK(*NO)

            STRCMTCTL  LCKLVL(*CS)
 FIRSTCALL: CALL       PGM(&APPLIB/STKLVLMTR)

ENDIT:      ENDPGM
```

STKLVLMTR: Stock Level Transaction Program

```
 PROCESS NOTRUNC NORANGE.
 IDENTIFICATION DIVISION.

*
*  Stock Level Program        (Transaction 5 (PF6 of TPCC))
*
*     Files Accessed:   - DSTRCT    (Read)
*                         ORDLIN    (Read)
*                         STOCK     (Read)
*
*     Views Used:       - ORDERLINLF (C1 - SQL)
*                         STOCKLF3   (C1 - SQL)
*
*     Join Position 1:    ORDLIN
*     Join Position 2:    STOCK
*
*   This program retrieves the number of unique items ordered
*   in the last 20 orders that have a stock quantity less than
*   the threshold level entered.  The threshold will be between
*   10 and 20.
*
*   The home Warehouse ID and Library of the data queues are
*   passed to this program as parameters.  All jobs processed by
*   this program are received from a data queue and the results
*   sent out using a returning data queue.  One Stock Level
*   program operates for 10 districts (ATPCCs) of a warehouse;
*   therefore, only the district value and threshold are
*   reinitialized with each job processed.  A threshold value of
*   99 will terminate Stock Level.
*
*   Processing the Stock Level transactions consists of:
*   1.  Read ICF file and wait for an entry
*   2.  Reading the DSTRCT file
*   3.  Calculating the order range
*   4.  Read the ORDERLINE and STOCK files (Avg of 200 reads)
*   5.  Count the number of records selected.
*   6.  Write to ICF file to return the answer to the user
*
    PROGRAM-ID. STKLVL.
    AUTHOR. DEPT-53G.
    INSTALLATION. IBM-ROCHESTER.
    DATE-WRITTEN. 09-03-95.
    DATE-COMPILED. 12-14-92.
  ENVIRONMENT DIVISION.
  CONFIGURATION SECTION.
    SOURCE-COMPUTER. IBM-AS400.
    OBJECT-COMPUTER. IBM-AS400.
    INPUT-OUTPUT SECTION.
    FILE-CONTROL.
      SELECT DSTRCT
             ASSIGN TO DATABASE-DSTRCT
             ORGANIZATION IS INDEXED
             ACCESS MODE IS RANDOM
             RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
             WITH DUPLICATES
             FILE STATUS IS DISTRICT-FILE-STATUS.
      SELECT ORDERLINE
             ASSIGN TO DATABASE-ORDLINLF
             ORGANIZATION IS INDEXED
             ACCESS MODE IS DYNAMIC
             FILE STATUS IS ORDERLINE-FILE-STATUS
             RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
             WITH DUPLICATES.
      SELECT STOCK
             ASSIGN TO DATABASE-STOCK
```

```
           ORGANIZATION IS INDEXED
           ACCESS MODE IS RANDOM
           FILE STATUS IS STOCK-FILE-STATUS
           RECORD KEY IS EXTERNALLY-DESCRIBED-KEY
           WITH DUPLICATES.
     SELECT STOCICFF ASSIGN TO WORKSTATION-STOCICFF-SI
              ORGANIZATION IS TRANSACTION
              FILE STATUS IS STATUS-IND MAJ-MIN.
   I-O-CONTROL.
     COMMITMENT CONTROL FOR DSTRCT ORDERLINE.
*
    DATA DIVISION.
    FILE SECTION.
 FD  DSTRCT
     LABEL RECORDS ARE STANDARD
     DATA RECORD IS DSTRCT-RECORD.
 01  DSTRCT-RECORD.
     COPY DDS-DSRCD OF DSTRCT.
 FD  ORDERLINE
     LABEL RECORDS ARE STANDARD
     DATA RECORD IS ORDERLINE-RECORD.
 01  ORDERLINE-RECORD.
     COPY DDS-OLRCD OF ORDLINLF.
 FD  STOCK
     LABEL RECORDS ARE STANDARD
     DATA RECORD IS STOCK-RECORD.
 01  STOCK-RECORD.
     COPY DDS-STRCD OF STOCK.
   FD STOCICFF
     LABEL RECORDS ARE STANDARD.
   01  ICFREC.
     COPY DDS-ALL-FORMATS OF STOCICFF.

    05 STKLVL-O REDEFINES STOCICFF-RECORD.
        06 FILLER                  PIC X(2).
        06 BLWSTK                  PIC S9(2) COMP-4.


**************************
 WORKING-STORAGE SECTION.
**************************
 01 STOCK-KEY.
    03 STOCK-KEY-DATA OCCURS 5 TIMES.
       05  KNAM   PIC X(10).
       05  KVAL   PIC S9(9) COMP-4.

 01 STOCK-HASH-PTR   USAGE POINTER.

 01 STOCK-HASH-NAME  PIC X(10).
 01 FUNCT            PIC X(1).
 01 KEYS             PIC S9(9) COMP-4.
 01 RET-CODE         PIC S9(9) COMP-4.

 01  RET-CODE-PTR    USAGE POINTER.
 01  STOCK-DEF-PTR   USAGE POINTER.
 01 STOCK-KEY-PTR USAGE POINTER.


 01 UPDATER                PIC X(1) VALUE "3".
 01 FETCHR                 PIC X(1) VALUE "1".
 01 FETCHUPDATE            PIC X(1) VALUE "2".

 01 CUST-KEYS              PIC 9(1) COMP-4 VALUE 3.
 01 STOCK-KEYS             PIC 9(1) COMP-4 VALUE 2.
 01 ITEM-KEYS              PIC 9(1) COMP-4 VALUE 1.


 01 MAJ-MIN.
    03  MAJ    PIC X(2).
    03  MIN    PIC X(2).

 01 CMNF-INDIC-AREA.
    03 CMNF-INDIC     PIC 1 OCCURS 99 TIMES
                      INDICATOR 1.

  01   TRANSACTION-INPUT.
         06    TXN-TYPE           PIC X.
         06    JOBNAME            PIC X(10).
         06    CLIENT-INPUT       PIC X(202).
         06 STKLVL-I REDEFINES CLIENT-INPUT.
           08 WID                 PIC S9(4) COMP-4.
           08 DID                 PIC S9(2) COMP-4.
           08 THRESHOLD           PIC S9(2) COMP-4.
           08 BLWSTK              PIC S9(4) COMP-4.

 77 STATUS-IND PIC X(2).
 01  ITEM-TABLE.
     05  ITEM-TABLE-ELEMENT OCCURS 300 TIMES.
         07  ITEM-TABLE-KEY.
             09  ITEM-TABLE-IID       PIC S9(6).

 77  DISTRICT-FILE-STATUS     PIC X(2).
 77  ORDERLINE-FILE-STATUS    PIC X(2).
 77  STOCK-FILE-STATUS        PIC X(2).
*
 01  NUMBER-OF-ITEMS          PIC S9(5) COMP-4.
 77 DUPS              PIC S9(5) COMP-4.
 77 HIORDER           PIC S9(9) COMP-4.
 77 LOORDER           PIC S9(9) COMP-4.
 77 IN-LOOP           PIC S9(5) COMP-4.
 77 OUT-LOOP          PIC S9(5) COMP-4.

*******************
 LINKAGE SECTION.
*******************

 PROCEDURE DIVISION.

*******************
 START-OF-PROGRAM.
*******************

 STOCK-WATCH-PROGRAM.

        OPEN INPUT DSTRCT
             INPUT STOCK
             INPUT ORDERLINE
        OPEN I-O STOCICFF

     READ STOCICFF
     INTO TRANSACTION-INPUT INDICATORS ARE CMNF-INDIC-AREA.
     IF MAJ = "81"
         GO TO END-OF-PROGRAM
     END-IF.


     MOVE "STOCK"            TO STOCK-HASH-NAME.
     MOVE "STWID"            TO KNAM OF STOCK-KEY-DATA(1).
     MOVE "STIID"            TO KNAM OF STOCK-KEY-DATA(2).

     SET STOCK-HASH-PTR  TO ADDRESS OF STOCK-HASH-NAME.
     SET RET-CODE-PTR    TO ADDRESS OF RET-CODE.
     SET STOCK-DEF-PTR   TO ADDRESS OF STOCK-RECORD.
     SET STOCK-KEY-PTR   TO ADDRESS OF STOCK-KEY.

 LOOP-FOREVER.

     MOVE WID OF STKLVL-I TO DWID, OLWID, STWID.

     MOVE DID OF STKLVL-I TO DID OF DSRCD, OLDID.

     READ DSTRCT.

*  Select all Detail Order Lines for the Last 20 Orders
*  that are below the threshold and also match a unique
*  stock item.

     SUBTRACT 1  FROM DNXTOR GIVING HIORDER.
     SUBTRACT 20 FROM DNXTOR GIVING LOORDER.
 READ-ORDERLINE-FILE.

     MOVE LOORDER  TO OLOID OF ORDERLINE-RECORD.

     MOVE 0 TO NUMBER-OF-ITEMS.

     READ ORDERLINE.

     MOVE OLIID TO STIID.

     MOVE WID OF STKLVL-I    TO KVAL OF STOCK-KEY-DATA(1)
     MOVE STIID              TO KVAL OF STOCK-KEY-DATA(2)

      CALL "qdbrunha" USING BY VALUE
           STOCK-HASH-PTR FETCHR STOCK-KEYS STOCK-KEY-PTR
           STOCK-DEF-PTR RET-CODE-PTR
     IF STQTY IS LESS THAN THRESHOLD
         ADD 1 TO NUMBER-OF-ITEMS
         MOVE STIID TO ITEM-TABLE-IID(NUMBER-OF-ITEMS)
     END-IF.

 READ-ORDERLINE-FILE-NEXT.

     READ ORDERLINE NEXT AT END GO TO SORT-ITEM-TABLE
     END-READ

     IF OLDID NOT EQUAL TO DID OF STKLVL-I
        GO TO SORT-ITEM-TABLE
     END-IF

     IF (OLOID > HIORDER) THEN GO TO SORT-ITEM-TABLE
        ELSE
          MOVE OLIID TO STIID

     MOVE WID OF STKLVL-I    TO KVAL OF STOCK-KEY-DATA(1)
     MOVE STIID              TO KVAL OF STOCK-KEY-DATA(2)

      CALL "qdbrunha" USING BY VALUE
           STOCK-HASH-PTR FETCHR STOCK-KEYS STOCK-KEY-PTR
           STOCK-DEF-PTR RET-CODE-PTR
         IF STQTY IS LESS THAN THRESHOLD
           ADD 1 TO NUMBER-OF-ITEMS
           MOVE STIID TO ITEM-TABLE-IID(NUMBER-OF-ITEMS)
         END-IF
     END-IF

     GO TO READ-ORDERLINE-FILE-NEXT.

 SORT-ITEM-TABLE.
       COMMIT

     PERFORM DISTINCT-SORT.

 DISPLAY-ANSWER.
     WRITE ICFREC FORMAT IS "SNDDATA"
           INDICATORS ARE CMNF-INDIC-AREA.

     READ STOCICFF
     INTO TRANSACTION-INPUT INDICATORS ARE CMNF-INDIC-AREA.
     IF MAJ = "81"
         GO TO END-OF-PROGRAM
     END-IF.

     GO TO LOOP-FOREVER.

 DISTINCT-SORT.
     IF NUMBER-OF-ITEMS > 0
       MOVE 1 TO BLWSTK OF STKLVL-O
       PERFORM VARYING OUT-LOOP FROM 2 BY 1 UNTIL OUT-LOOP
           IS GREATER THAN NUMBER-OF-ITEMS
         MOVE 0 TO DUPS
         PERFORM VARYING IN-LOOP FROM 1 BY 1 UNTIL IN-LOOP
             IS EQUAL TO OUT-LOOP
           IF ITEM-TABLE-ELEMENT(IN-LOOP) =
             ITEM-TABLE-ELEMENT(OUT-LOOP)
             ADD 1 TO DUPS
             END-IF
           END-PERFORM
         IF DUPS = 0
           ADD 1 TO BLWSTK OF STKLVL-O
           END-IF
       END-PERFORM
       ELSE MOVE 0 TO BLWSTK OF STKLVL-O.

 END-OF-PROGRAM.

     STOP RUN.
     EXIT.
```

DLVRICFF: Delivery ICF File on Server

```
A*************************************************************
A*                                                           *
A*           ICF file for NEW ORDER transaction              *
A*                                                           *
A*************************************************************
A*   File level indicators:
A*
A                                     INDARA
A*
A*************************************************************
A*            ICF RECORD FORMATS                             *
A*************************************************************
A          R RCVDATA                  TEXT('DATA RECEIVED FROM -
A                                     THE REMOTE SYSTEM.')
A            RCVFLD       230A       TEXT('RECORD KEY')
A          R SNDDATA                  TEXT('DATA SENT TO THE   -
A                                     REMOTE SYSTEM.')
```

```
A            SNDFLD       670A         TEXT('RECORD KEY')
A          R EVOKPGM                   TEXT('EVOKE THE TARGET   -
A                                      PROGRAM.')
A  50                                  EVOKE(&LIB/&PGMID)
A                                      SECURITY(3 *USER)
A            PGMID         10A  P
A            LIB           10A  P


A          R ENDREC
```

MAINICFF: Neworder / Payment / Order Status ICF Files on Server

```
A*****************************************************************
A*                                                               *
A*            ICF file for NEW ORDER transaction                 *
A*                                                               *
A*****************************************************************
A*   File level indicators:
A*
A                                      INDARA
A*
A                                      RCVTRNRND(15 'END OF DATA')
A*
A  30                                  DETACH
A*
A                                      INDTXT(30 '30->DETACH TARG-
A                                      ET PROGRAM.')
A*
A                                      RCVDETACH(35 'RECEIVED    -
A                                      DETACHED.')
A*
A*
A*****************************************************************
A*           ICF RECORD FORMATS                                  *
A*****************************************************************
A          R RCVDATA                   TEXT('DATA RECEIVED FROM -
A                                      THE REMOTE SYSTEM.')
A            RCVFLD       230A         TEXT('RECORD KEY')
A          R SNDDATA                   TEXT('DATA SENT TO THE   -
A                                      REMOTE SYSTEM.')
A            SNDFLD       670A         TEXT('RECORD KEY')
A          R EVOKPGM                   TEXT('EVOKE THE TARGET   -
A                                      PROGRAM.')
A                                      SECURITY(3 *USER)
A                                      EVOKE(TPCCNCSRVR/ATPCCMTRA)


A          R ENDREC
```

STOCICFF: Stock Level ICF File on Server

```
A*****************************************************************
A*                                                               *
A*            ICF file for NEW ORDER transaction                 *
A*                                                               *
A*****************************************************************
A*   File level indicators:
A*
A                                      INDARA
A*
A                                      RCVTRNRND(15 'END OF DATA')
A*
A  30                                  DETACH
A*
A                                      INDTXT(30 '30->DETACH TARG-
A                                      ET PROGRAM.')
A*
A                                      RCVDETACH(35 'RECEIVED    -
A                                      DETACHED.')
A*
A*
A*****************************************************************
A*           ICF RECORD FORMATS                                  *
A*****************************************************************
A          R RCVDATA                   TEXT('DATA RECEIVED FROM -
A                                      THE REMOTE SYSTEM.')
A            RCVFLD       230A         TEXT('RECORD KEY')
A          R SNDDATA                   TEXT('DATA SENT TO THE   -
A                                      REMOTE SYSTEM.')
A            SNDFLD       670A         TEXT('RECORD KEY')
A          R EVOKPGM                   TEXT('EVOKE THE TARGET   -
A                                      PROGRAM.')
A                                      SECURITY(3 *USER)
A                                      EVOKE(TPCCNCSRVR/ATPCCMTRS)
A          R ENDREC
```

GETTIME: Get System Time

```
/*****************************************************************/
/*                                                               */
/* OCO Source Materials                                          */
/*                                                               */
/* The Source code for this program is not published or otherwise */
/* divested of its trade secrets, irrespective of what has been  */
/* deposited with the U.S. Copyright office                      */
/*                                                               */
/* TPCCCLIENT LIBRARY                                            */
/* (C) Copyright IBM Corp. 1997,1998                             */
/*                                                               */
/*****************************************************************/

#pragma comment (copyright, \
"TPCCCLIENT LIBRARY\
(C) Copyright IBM Corp. 1997,1998.\
All rights reserved.\
US Government Users Restricted Rights-\
Use, duplication or disclosure restricted \
by GSA ADP Schedule Contract with IBM Corp.\
Licensed Materials-Property of IBM")

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <leawi.h>
#include <QSYSINC/MIH/MATTOD>
#include <qwcrdtaa.h>
#define BASE_OFFSET 4503599627.370496
#define OFFSET     109142676000.0 /* set to a fixed value */
/* Define the time range flag */
#define HIGH_FLAG 'B'
/* #include <CEEDATM> */
struct dateTime {
   _MI_Time returnTimestamp;
 };
void gettime(struct dateTime * new_date_time)
{
   mattod(new_date_time->returnTimestamp);/* get the time of day */
   new_date_time->returnTimestamp[7] = HIGH_FLAG;/* set the high flag */
}

void gettime_string(char * timein, char* timestamp )
{
  double converted = 0;/* the output value */
  double seconds;/* number of seconds in floting point since 14 October 1582 */
  double scale=0;/* scae factor from MI_Time to seconds */
  double second = 16000000;/* number of MI units in one second */
  double offset=/*number of seconds from MI_base to seconds base */
         OFFSET;
         /* note this has a one hour offset for daylight time */
         /* base date is Aug 23, 1928 noon */
         /* the daylight time offset gives 1:00 PM time */
  double base =1;/* the base for the value used */
  double Base_sec = 0;/* the base for the value used */
  char picture[] = "Mmm DD, YYYY HH:MI:SS AP.";/* timestamp defintion */

  _FEEDBACK fb;/* feedback value */
  int count;/* loop counter */
  char high_flag;
  high_flag = timein[7];/* get the high byte flag */
  if ( high_flag != 'B' )
  {
    if (( high_flag == 'A' )&(( timein[0]&&0xF0 )> 0x70))
    /* first part of the A date range this section is smaller then */
    /* the base section                                            */
    {
       offset = OFFSET - BASE_OFFSET; /* next lower range */
    }
    else
    {
       if (( high_flag == 'C' )&(( timein[0]&&0xF0 )< 0x80))
    /* second part of the C date range this section is larger then */
       {
          offset = OFFSET + BASE_OFFSET; /* next higher range */
       }
    }
  }
  for ( count = 0; count <7; count++ )
  {
     converted = converted + ((timein[6-count]&255) * base);
     base = base * 256;
  }
  /* printf ("second is %f\n", second ); */
  seconds = (converted/second) + offset;
  memset (timestamp, ' ', 27 );/* reset time stamp */
  CEEDATM ( &seconds,&picture[0],timestamp,&fb);/* convert to timestamp */
  timestamp[26] = '\0';/* terminate the time stamp */
}
void printtime(char * timein )
{
  char time_stamp[27];/* storage for text */
  gettime_string(timein, time_stamp );/* get the time stamp */
  printf ( "TIME: %s\n", time_stamp );/* print output */
}
```

GETDTTMSTR: Get Date Time String

```
/*****************************************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stddef.h>
#include <math.h>
#include <ctype.h>
#include <decimal.h>
#include <recio.h>
#include <errno.h>
#include <time.h>
#include <xxfdbk.h>
#include <xxcvt.h>
#include <micomput.h>
#include <QSYSINC/MIH/MIDTTM>
#include <qsysinc/mih/cvtch>
/*****************************************************************************/
/* Function Prototypes                                                       */
/*****************************************************************************/
void MI_Template (_INST_Template_T1 **, _DDAT_T **, _DDAT_T **, char );

/*****************************************************************************/
/* Date and time coversion pointers                                          */
/*****************************************************************************/
 _INST_Template_T1 *inst_t1;          /* MI template         */
 _DDAT_T   *ddat_t1 , *ddat_t2 ;      /* DDAT pointers       */
 /* Internal format for end of Gregorian timeline: Jan. 1,  0001 */
 #define GREGORIAN_TIMELINE_START 1721424
 /* Internal format for end of Gregorian timeline: Jan. 1, 10000 */
 #define GREGORIAN_TIMELINE_END   5373485
/*****************************************************************************/
/*        Start                                                              */
/*****************************************************************************/
getdttmstr(char * timein, char* timestamp, char type)
{

 char tmstamp[27];
 static char first = 'Y';

 if (( timein[7] == ' ' )||(timein[7]=='0'))
```

```
{
switch ( type )
{
   case 'd':
      memset ( timestamp, '0', 8 );/* set the null date */
   break;
   case 't':
      memset ( timestamp, '0', 6 );/* set the null date */
   break;
   case 'h':
      memset ( timestamp, '0', 8 );/* set the null date */
   break;
   case 'b':
      memset ( timestamp, '0', 14 );/* set the null date */
   break;
   case 'l':
      memcpy( &timestamp[0],/* copy hour min sec */
      "0000-00-00.00.00.00.000000\0", 27 );
   break;
   case 'm':
      memcpy( &timestamp[0],/* copy hour min sec */
      "0000-00-00 00:00:00.000000\0", 27 );/* set null string */
   break;
   case 'n':
      memcpy( &timestamp[0],/* copy hour min sec */
      "0000/00/00 00:00:00.000000\0", 27 );/* set null string */
   break;
   default:
   break;
  } /* end of switch */
}
else
{
MI_Template(&inst_t1,&ddat_t1,&ddat_t2, 'P');
timein[7] = 0;/* clear the era flag */
cvtts(tmstamp, timein, inst_t1);

switch ( type )
{
   case 'd':
      memcpy( &timestamp[4], &tmstamp[0], 4 ); /* copy the year */
      memcpy( &timestamp[0], &tmstamp[5], 2 ); /* copy month    */
      memcpy( &timestamp[2], &tmstamp[8], 2 ); /* copy day      */
   break;
   case 't':
   case 'h':
      memcpy( &timestamp[0], &tmstamp[11], 2 ); /* copy hour     */
      memcpy( &timestamp[2], &tmstamp[14], 2 );/* copy minuites */
      memcpy( &timestamp[4], &tmstamp[17], 2 ); /* copy secound  */
      if ( type == 'h')
      memcpy( &timestamp[6], &tmstamp[20], 2 ); /* copy secound  */
   break;
   case 'b':
      memcpy( &timestamp[4], &tmstamp[0], 4 ); /* copy the year */
      memcpy( &timestamp[0], &tmstamp[5], 2 ); /* copy month    */
      memcpy( &timestamp[2], &tmstamp[8], 2 ); /* copy days     */
      memcpy( &timestamp[8], &tmstamp[11], 2 ); /* copy hour     */
      memcpy( &timestamp[10], &tmstamp[14], 2 );/* copy minuites */
      memcpy( &timestamp[12], &tmstamp[17], 2 ); /* copy secound  */
   break;
   case 'l':
      memcpy( &timestamp[0], &tmstamp[0], 27 );/* copy hour min sec */
   break;
   case 'm':
   case 'n':
      memcpy(&timestamp[0],&tmstamp[5],2);       /* get month    */
      memcpy(&timestamp[3],&tmstamp[8],2);       /* get day      */
      memcpy(&timestamp[6],&tmstamp[0],4);       /* get year     */
      memcpy(&timestamp[11],&tmstamp[11],2);     /* get hours    */
      memcpy(&timestamp[14],&tmstamp[14],2);     /* get minutes  */
      memcpy(&timestamp[17],&tmstamp[17],2);     /* get seconds  */
      timestamp[10] = ' ';/* set the time seperator */
      timestamp[13] = ':';/* set the time seperator */
      timestamp[16] = ':';/* set the time seperator */
      if ( type == 'n' )
      {
        timestamp[2] = '/';/* set the date seperator */
        timestamp[5] = '/';/* set the date seperator */
      }
      else
      {
        timestamp[2] = '-';/* set the date seperator */
        timestamp[5] = '-';/* set the date seperator */
      }
   break;
   default:
   break;
 } /* end of switch */
 } /* end of else   */
}
void makemitime ( char * date, char *time, char * flag,
                  char * output )
{
  char working[27];/* the working 26 character data input */
  char cvt_type = 'I';
  char impi_time[9] = {0x7D,0x66,0x27,0x9F,0xE4,0xF5,0x00,0x00};
 char tmstamp1[27] = "1998-07-24-13.16.55.000000";

  /* format   YYYY-MM-DDHH.MI.SS.999999   */
 /* location  01234567890123456789012345  */
 /* tens      0         1         2       */
  memcpy ( &working[0], &date[4], 4); /* get the year */
  memcpy ( &working[5], &date[0], 2); /* get the month */
  memcpy ( &working[8], &date[2], 2); /* get the day   */
  memcpy ( &working[11], &time[0], 2); /* get the hour  */
  memcpy ( &working[14], &time[2], 2); /* get the minite */
  memcpy ( &working[17], &time[4], 2); /* get the secound */
  working[4] = '-'; /* set the dashes */
  working[7] = '-'; /* set the dashes */
  working[10] = '-'; /* set the dashes */
  working[13] = '.'; /* set the time seperator */
  working[16] = '.';/* set the porid */
  working[19] = '.';/* set the porid */
  working[26] = '\0';/* terminate the string */
  if ( flag[0] == 'H' )
  {
        memset ( &working[20], '9', 6 );/* set the value to the maximume */
  }
  else
  {
        memset ( &working[20], '0', 6 );/* set the value to the minimume */
  }
    MI_Template(&inst_t1,&ddat_t1,&ddat_t2, cvt_type);
    /* printf ( "The time  stamp was %s\n", tmstamp1 ); */
       printf ( "The time is         %s\n", working );
   cvtts( output, working, inst_t1);
}
 /****************** MI_Template ***********************************/
 /* Function: MI_Template                                          */
 /*                                                                */
 /* Description: Allocate and fill in MI instruction template used */
 /*              in date and time conversions.                     */
 /*                                                                */
 /******************************************************************/
```

```
void  MI_Template(_INST_Template_T1 **inst_t, _DDAT_T **ddat1,
                  _DDAT_T **ddat2, char cvt_type)
{
 static char first = 'Y';
 /******************************************************************/
 /* Date and Time conversion (mi template)                         */
 /******************************************************************/
  _Era_Table_T       *era_t1,  *era_t2; /* Era table pointers          */
  _Calendar_Table_T  *cal_t1,  *cal_t2; /* Calendar table pointers     */

 int                DDAT_Length, Calendar_Offset;
 int                DDAT_Size, Template_Size;
 int                DDAT_Offset1, DDAT_Offset2;
 /******************************************************************/
 if (first == 'Y')             /* First call ?                 */
 {                             /* Yes, allocate template etc.  */
    first ='N';

 DDAT_Length = sizeof(_DDAT_T)+2*(sizeof(_Calendar_Table_T))
               -sizeof(short) + sizeof(_Era_Table_T)-1;

 Calendar_Offset = offsetof(_DDAT_T, Tables)+sizeof(_Era_Table_T);

 DDAT_Size = 2*DDAT_Length +
             2*(sizeof(int)) + /* DDAT_Offset  */
             10 +              /* reserved4    */
             sizeof(short) +   /* Num_DDATS    */
             sizeof(int);      /* DDAT_Size    */
                               /* DDAT offset  */

 Template_Size = 2*DDAT_Length+offsetof(_INST_Template_T1,DDAT)
                 + sizeof(int);
                               /* DDAT1 offset */
 DDAT_Offset1  = (offsetof(_INST_Template_T1,DDAT) -
                 offsetof(_INST_Template_T1,DDAT_Size)) + sizeof(int);
                               /* DDAT2 offset */

 DDAT_Offset2  = (offsetof(_INST_Template_T1,DDAT) -
                 offsetof(_INST_Template_T1,DDAT_Size))+sizeof(int)+DDAT_Length;

    *inst_t        = (_INST_Template_T1 *)malloc(Template_Size);

 /* Fill in Instruction Template */


 memset(*inst_t,'\0', sizeof(_INST_Template_T1));
 (* inst_t)->Template_Size= Template_Size; /* Instruction template size */
 (* inst_t)->DDAT_1       = 1;             /* Operand 1 DDAT        */
 (* inst_t)->DDAT_2       = 2;             /* Operand 2 DDAT        */
 (* inst_t)->Length_1     = 26;            /* Result timestamp      */
 (* inst_t)->Length_2     = 8;             /* input mitime          */
 (* inst_t)->DDAT_Size       = DDAT_Size;  /* Size of DDAT list     */
 (* inst_t)->Num_DDATs       = 2;          /* Number of DDATs       */
 (* inst_t)->DDAT_Offset[0]  = DDAT_Offset1; /* DDAT offset1        */
 *((* inst_t)->DDAT_Offset+1)= DDAT_Offset2; /* DDAT offset2        */

 *ddat1   = (_DDAT_T *)&((* inst_t)->DDAT_Offset+2);
 era_t1 = (_Era_Table_T *)&((* ddat1)->Tables); /* set era table ptr */
 cal_t1    = (_Calendar_Table_T *)((char *)*ddat1 + Calendar_Offset);
 *ddat2   = (_DDAT_T *)((char *)*ddat1 + DDAT_Length);
 era_t2    = (_Era_Table_T *)&((* ddat2)->Tables);
 cal_t2    = (_Calendar_Table_T *)((char *)*ddat2 + Calendar_Offset);

 /* Fill in DDAT1 */

 (* ddat1)->DDAT_Length = DDAT_Length;      /* Set DDAT length       */
 (* ddat1)->Format_Code =_SAA_TIMESTAMP; /* Set result format code     */
 (* ddat1)->Hour_Zone = 24;            /* set target hours            */
 (* ddat1)->Min_Zone  = 60;            /* set target minutes          */
 (* ddat1)->Calendar_Offset=Calendar_Offset;/* Calendar offset       */
 memset ((* ddat1)->reserved, 0, 6 );  /* set the reserved values    */
 /* Fill in Era Table for DDAT1 */

    era_t1->Num_Elems = 1;               /* Set number of era elements  */
    era_t1->Element[0].Origin_Date = GREGORIAN_TIMELINE_START;
    era_t1->Element[0].Era_Name[0]= 'A'; /* Set era name, to ....       */
    era_t1->Element[0].Era_Name[1]= 'D'; /* AD                          */
    memset (era_t1->Element[0].reserved, 0, 12 );/* set the reserved v alues */


 /* Fill in Calendar Table for DDAT1 */

    cal_t1->Num_Elems = 2;               /* Set Calendar elements       */
    cal_t1->Element->Effect_Date     = GREGORIAN_TIMELINE_START;
    cal_t1->Element->Type            = 0x0001; /* Set element type      */
    memset(cal_t1->Element->reserved,'\0',10); /* Initialize reserved   */
    (cal_t1->Element+1)->Effect_Date = GREGORIAN_TIMELINE_END;
    (cal_t1->Element+1)->Type = 0;
    memset((cal_t1->Element+1)->reserved,'\0',10);

 /* Fill in DDAT2 */

 (* ddat2)->DDAT_Length = DDAT_Length;     /* Set DDAT length        */
 (* ddat2)->Format_Code =_IMPI_CLOCK;  /* Set source format code      */
 (* ddat2)->Hour_Zone = 24;           /* set target hours             */
 (* ddat2)->Min_Zone  = 60;           /* set target minutes           */
 (* ddat2)->Calendar_Offset=Calendar_Offset;/* Calendar offset       */
 (* ddat2)->Calendar_Offset = Calendar_Offset;
 memset ((* ddat2)->reserved, 0, 6 );/* set the reserved values   */

 /* Fill in Era Table for DDAT2 */

    era_t2->Num_Elems = 1;
    era_t2->Element[0].Origin_Date   = GREGORIAN_TIMELINE_START;
    era_t2->Element[0].Era_Name[0]   = 'A';
    era_t2->Element[0].Era_Name[1]   = 'D';
    memset (era_t2->Element[0].reserved, 0, 12 );/* set the reserved v alues */

 /* Fill in Calendar Table for DDAT2 */

    cal_t2->Num_Elems               = 2;
    cal_t2->Element->Effect_Date     = GREGORIAN_TIMELINE_START;
    cal_t2->Element->Type            = 0x0001;
    memset(cal_t2->Element->reserved,'\0',10);
    (cal_t2->Element+1)->Effect_Date = GREGORIAN_TIMELINE_END;
    (cal_t2->Element+1)->Type        = 0;
    memset((cal_t2->Element+1)->reserved,'\0',10);
 }/* end of if (first) */
 if (cvt_type == 'I')                        /* converting to mitime ?  */
 {                                           /* Yes, then ....          */
    (* ddat1)->Format_Code =_IMPI_CLOCK;     /* Set source format code  */
    (* ddat2)->Format_Code =_SAA_TIMESTAMP;  /* Set source format code  */
    (* inst_t)->Length_1     = 8;            /* Result timestamp        */
    (* inst_t)->Length_2     = 26;           /* input mitime            */
 }
 else
 {
    (* ddat1)->Format_Code =_SAA_TIMESTAMP;  /* Set source format code  */
    (* ddat2)->Format_Code =_IMPI_CLOCK;     /* Set source format code  */
    (* inst_t)->Length_1     = 26;           /* Result timestamp        */
    (* inst_t)->Length_2     = 8;            /* input mitime            */
 }
}/* end of proceedure */
```

Appendix E. RTE Scripts

RTE Parameters_1

```
/* NOTICE! Variables can be a maximum of 200 characters in length */

/*****************/
/* Hosts section */
/*****************/
MASTER slavmstr
SLAVES
rte8,rte9,rte10,rte11,rte12,rte13,rte14,rte15,rte16,rte19,rte42,rte55,rte56,rte58,rte61,rte62,
rte63,rte64,rte65,rte81,rte100,rte101,rte102,rte103,rte104,rte105,rte106,rte107,rte108


/******************/
/* Reports section */
/******************/
/******          USE ANALYZE2 COMMAND AFTER THE RUN TO GENERATE REPORTS         ************/


/************************/
/* Client Login section */
/************************/
CLIENT TILLY    tpccuser pete96july
CLIENT DUCHESS  tpccuser pete96july
CLIENT MANDY    tpccuser pete96july
CLIENT NELLIE   tpccuser pete96july
CLIENT VERNIE   tpccuser pete96july
CLIENT SETH     tpccuser pete96july
CLIENT MOLLIE   tpccuser pete96july
CLIENT KIRBY    tpccuser pete96july
CLIENT FRITZIE  tpccuser pete96july
CLIENT MICKY    tpccuser pete96july
CLIENT LUCKIE   tpccuser pete96july
CLIENT LUCI     tpccuser pete96july
CLIENT PATCHIE  tpccuser pete96july
CLIENT PEGGIE   tpccuser pete96july
CLIENT CHEWY    tpccuser pete96july
CLIENT WINNIE   tpccuser pete96july
CLIENT CHESTER  tpccuser pete96july
CLIENT HUNTER   tpccuser pete96july
CLIENT BABI     tpccuser pete96july
CLIENT BRADI    tpccuser pete96july
CLIENT HORSEFLY tpccuser pete96july


/***************************************/
/* Terminal Emulation Program section */
/***************************************/
PROGRAM me5250 "/usr/bin/me5250 %s 2>&1"      /* Without SNA */
PROGRAM me5250s "/usr/bin/me5250 -s %s 2>&1"  /* With SNA    */

/***********************/
/* Sub-network section */
/***********************/
/* Specify an emulator program and a group of slaves to use together */
SUB_NETWORK i06t me5250  rte55,rte56,rte58,rte61,rte62,rte63,rte64,rte65
SUB_NETWORK i07t me5250  rte8,rte14,rte15,rte16,rte19
SUB_NETWORK i08t me5250  rte9,rte10,rte11,rte12,rte13
SUB_NETWORK i09t me5250  rte42
SUB_NETWORK i10t me5250  rte100
SUB_NETWORK i11t me5250  rte109
SUB_NETWORK i12t me5250  rte102
SUB_NETWORK i13t me5250  rte81
SUB_NETWORK i14t me5250  rte103
SUB_NETWORK i15t me5250  rte104
SUB_NETWORK i16t me5250  rte106
SUB_NETWORK i17t me5250  rte107
SUB_NETWORK i18t me5250  rte108
SUB_NETWORK i19t me5250  rte105
SUB_NETWORK i20t me5250  rte101

/*********************/
/* Start run section */
/*********************/
START BRADI     i06t 4000
START BABI      i07t 1780
START HORSEFLY i08t 1780
START CHEWY     i09t 370
START FRITZIE  i10t 370
/*START HUNTER   i11t 370*/
START KIRBY     i12t 370
START LUCI      i13t 370
START LUCKIE    i14t 370
START MICKY     i15t 370
START MOLLIE    i16t 370
START PATCHIE   i17t 370
START PEGGIE    i18t 370
START WINNIE    i19t 370
START CHESTER   i20t 370

/***********************/
/* Run Length section */
/***********************/
RAMPUP    = 40:00
RUNTIME   = 1:30:00
RAMPDOWN = 30:00
INTERVAL = 5:00   /* Interval to calculate mix from */

/*****************************/
/* Global Variable section */
/*****************************/
NOBEGIN               = 1
KEYSTROKE_PACKET_SIZE = 10

MAX_CONCURRENT_SPAWN  = 1
SPAWN_COUNT           = 1
LOGIN_TIMEOUT         = 120 /* seconds */
BEGIN_WAIT            = 1200

/*****************/
/* Flags section */
/*****************/
#define TES_FLAG_TRACE          0x00000010
#define TES_FLAG_KEYSTROKE_TIME 0x00000200
#define TES_FLAG_LOCAL_LOG      0x00000400
#define TES_FLAG_LOCAL_TRACE    0x00000800

SETFLAG ALL TES_FLAG_TRACE
SETFLAG ALL TES_FLAG_LOCAL_TRACE

/*********************************/
/* Transaction Settings section */
/*********************************/
/*          Think Time, Menu Delay, Resp Delay, %desired, % min, % max */
NEWORDER =    "12.20,     0.00,        0.00"
PAYMENT  =    "12.20,     0.00,        0.00,      43.2,    42.0,  45.3"
ORDSTAT  =    "10.20,     0.00,        0.00,       4.1,     3.9,   4.2"
DELIVERY =    "05.20,     0.00,        0.00,       4.1,     3.9,   4.2"
STOCKLEV =    "05.20,     0.00,        0.00,       4.1,     3.9,   4.2"
```

RTE Parameters_2

```
/* NOTICE! Variables can be a maximum of 200 characters in length */

/*****************/
/* Hosts section */
/*****************/
MASTER rte41
SLAVES
rte122,rte28,rte68,rte69,rte70,rte71,rte72,rte73,rte74,rte75,rte76,rte77,rte78,rte79,rte80,rte
113,rte114,rte116,rte117,rte119,rte123,rte124,rte126,rte127


/******************/
/* Reports section */
/******************/
/******         USE ANALYZE2 COMMAND AFTER THE RUN TO GENERATE REPORTS         ************/


/************************/
/* Client Login section */
/************************/
CLIENT TILLY    tpccuser pete96july
CLIENT DUCHESS  tpccuser pete96july
CLIENT MANDY    tpccuser pete96july
CLIENT NELLIE   tpccuser pete96july
CLIENT VERNIE   tpccuser pete96july
CLIENT SETH     tpccuser pete96july
CLIENT MOLLIE   tpccuser pete96july
CLIENT KIRBY    tpccuser pete96july
CLIENT FRITZIE  tpccuser pete96july
CLIENT MICKY    tpccuser pete96july
CLIENT LUCKIE   tpccuser pete96july
CLIENT LUCI     tpccuser pete96july
CLIENT PATCHIE  tpccuser pete96july
CLIENT PEGGIE   tpccuser pete96july
CLIENT CHEWY    tpccuser pete96july
CLIENT WINNIE   tpccuser pete96july
CLIENT CHESTER  tpccuser pete96july
CLIENT HUNTER   tpccuser pete96july
CLIENT BABI     tpccuser pete96july
CLIENT BRADI    tpccuser pete96july
CLIENT HORSEFLY tpccuser pete96july


/***************************************/
/* Terminal Emulation Program section */
/***************************************/
PROGRAM me5250 "/usr/bin/me5250 %s 2>&1"      /* Without SNA */
PROGRAM me5250s "/usr/bin/me5250 -s %s 2>&1"  /* With SNA    */

/***********************/
/* Sub-network section */
/***********************/
/* Specify an emulator program and a group of slaves to use together */
SUB_NETWORK i05t me5250  rte122,rte28,rte68,rte69,rte70,rte71,rte72,rte73
SUB_NETWORK i06t me5250  rte74,rte75,rte76,rte77,rte78,rte79,rte80,rte127
SUB_NETWORK i07t me5250  rte124,rte123,rte126,rte113,rte114,rte119,rte116,rte117

/*********************/
/* Start run section */
/*********************/
START SETH      i05t 4000
START TILLY     i06t 4000
START VERNIE    i07t 4000

/***********************/
/* Run Length section */
/***********************/
RAMPUP    = 40:00
RUNTIME   = 1:30:00
RAMPDOWN = 30:00
INTERVAL = 5:00   /* Interval to calculate mix from */

/*****************************/
/* Global Variable section */
/*****************************/
NOBEGIN               = 1
KEYSTROKE_PACKET_SIZE = 10

MAX_CONCURRENT_SPAWN  = 1
SPAWN_COUNT           = 1
LOGIN_TIMEOUT         = 120 /* seconds */
BEGIN_WAIT            = 1200

/*****************/
/* Flags section */
/*****************/
#define TES_FLAG_TRACE          0x00000010
#define TES_FLAG_KEYSTROKE_TIME 0x00000200
#define TES_FLAG_LOCAL_LOG      0x00000400
#define TES_FLAG_LOCAL_TRACE    0x00000800

SETFLAG ALL TES_FLAG_TRACE
SETFLAG ALL TES_FLAG_LOCAL_TRACE

/*********************************/
/* Transaction Settings section */
/*********************************/
/*          Think Time, Menu Delay, Resp Delay, %desired, % min, % max */
NEWORDER =    "12.20,     0.00,        0.00"
PAYMENT  =    "12.20,     0.00,        0.00,      43.2,    42.0,  45.3"
ORDSTAT  =    "10.20,     0.00,        0.00,       4.1,     3.9,   4.2"
DELIVERY =    "05.20,     0.00,        0.00,       4.1,     3.9,   4.2"
STOCKLEV =    "05.20,     0.00,        0.00,       4.1,     3.9,   4.2"
```

RTE Parameters_3

```
/* NOTICE! Variables can be a maximum of 200 characters in length */

/*****************/
/* Hosts section */
/*****************/
MASTER rte110
```

```
SLAVES
rte5,rte6,rte7,rte30,rte34,rte36,rte38,rte39,rte40,rte44,rte50,rte51,rte52,rte53,rte57,rte59,r
te60,rte111,rte112,rte118,rte120,rte121,rte125,rte129


/******************/
/* Reports section */
/******************/
/******         USE ANALYZE2 COMMAND AFTER THE RUN TO GENERATE REPORTS         *************/


/************************/
/* Client Login section */
/************************/
CLIENT TILLY     tpccuser pete96july
CLIENT DUCHESS   tpccuser pete96july
CLIENT MANDY     tpccuser pete96july
CLIENT NELLIE    tpccuser pete96july
CLIENT VERNIE    tpccuser pete96july
CLIENT SETH      tpccuser pete96july
CLIENT MOLLIE    tpccuser pete96july
CLIENT KIRBY     tpccuser pete96july
CLIENT FRITZIE   tpccuser pete96july
CLIENT MICKY     tpccuser pete96july
CLIENT LUCKIE    tpccuser pete96july
CLIENT LUCI      tpccuser pete96july
CLIENT PATCHIE   tpccuser pete96july
CLIENT PEGGIE    tpccuser pete96july
CLIENT CHEWY     tpccuser pete96july
CLIENT WINNIE    tpccuser pete96july
CLIENT CHESTER   tpccuser pete96july
CLIENT HUNTER    tpccuser pete96july
CLIENT ZOEY      tpccuser pete96july
CLIENT BABI      tpccuser pete96july
CLIENT BRADI     tpccuser pete96july
CLIENT HORSEFLY tpccuser pete96july

/*************************************/
/* Terminal Emulation Program section */
/*************************************/
PROGRAM me5250 "/usr/bin/me5250 %s 2>&1"      /* Without SNA */
PROGRAM me5250s "/usr/bin/me5250 -s %s 2>&1"  /* With SNA    */

/***********************/
/* Sub-network section */
/***********************/
/* Specify an emulator program and a group of slaves to use together */
SUB_NETWORK i01t me5250  rte5,rte6,rte7,rte125,rte112,rte118,rte120,rte111
SUB_NETWORK i02t me5250  rte30,rte57,rte59,rte60,rte34,rte36,rte38,rte39
SUB_NETWORK i03t me5250  rte40,rte129,rte121,rte44,rte50,rte51,rte52,rte53

/*********************/
/* Start run section */
/*********************/
START DUCHESS i01t 4000
START MANDY   i02t 4000
START NELLIE  i03t 4000
/***********************/
/* Run Length section */
/***********************/
RAMPUP   = 40:00
RUNTIME  = 1:30:00
RAMPDOWN = 30:00
INTERVAL = 5:00   /* Interval to calculate mix from */

/*****************************/
/* Global Variable section */
/*****************************/
NOBEGIN               = 1
KEYSTROKE_PACKET_SIZE = 10

MAX_CONCURRENT_SPAWN  = 1
SPAWN_COUNT           = 1
LOGIN_TIMEOUT         = 120 /* seconds */
BEGIN_WAIT            = 1200

/*****************/
/* Flags section */
/*****************/
#define TES_FLAG_TRACE          0x00000010
#define TES_FLAG_KEYSTROKE_TIME 0x00000200
#define TES_FLAG_LOCAL_LOG      0x00000400
#define TES_FLAG_LOCAL_TRACE    0x00000800

SETFLAG ALL TES_FLAG_TRACE
SETFLAG ALL TES_FLAG_LOCAL_TRACE

/*********************************/
/* Transaction Settings section */
/*********************************/
/*          Think Time, Menu Delay, Resp Delay, %desired, % min, % max */
NEWORDER =    "12.20,     0.00,        0.00"
PAYMENT  =    "12.20,     0.00,        0.00,      43.2,    42.0,  45.3"
ORDSTAT  =    "10.20,     0.00,        0.00,       4.1,     3.9,   4.2"
DELIVERY =    "05.20,     0.00,        0.00,       4.1,     3.9,   4.2"
STOCKLEV =    "05.20,     0.00,        0.00,       4.1,     3.9,   4.2"
```

Declarations

```
#ifndef USER_TPCC_H
#define USER_TPCC_H
/*****************************************************************************/
/***   run-time constant for customer last name from 0 to 255,          ***/
/***   run-time constant for customer id from 0 to 1023,                ***/
/***   run-time constant for item id from 0 to 8191.                    ***/
/*****************************************************************************/
#define        LASTC          117
#define        CUSTC          319
#define        ITEMC          3849

/*****************************************************************************/
/***   response type
                ***/
/*****************************************************************************/
/* #define OK  1  */
/* #define ERROR              -1 */

/*****************************************************************************/
/***   transaction type
               ***/
/*****************************************************************************/
#define        NEWORDER       1
#define        PAYMENT                     2
#define        ORDSTAT                     3
#define        DELIVERY       4
#define        STOCKLEV       5

/*****************************************************************************/
/***   transaction structures
***/
/*****************************************************************************/
struct neword_struct {
    char            invalid; /* transaction completed sucessfully */
    long            did;
    long            cid;
    long            oid;       /* Order-ID returned from client */
    long            nloop;   /* number of order line, avg = 15 */
    char            oremote; /* 1 for remote order, 10% */
    long            olremote;/* number of remote order line, 1% */
    char            rollback;/* actually saw rollback text on screen */
    struct items_struct {
              long            olswid;
              long            oliid;
              long            olquantity;
    } item[15];
};

struct payment_struct {
    char            invalid; /* transaction completed sucessfully */
    long did;
    long cid;
    long cwid;
    long cdid;
    char clast[17];
    double amount;
    char byname;                                        /* 1 for by last name, 0 for by id
*/
    char remote;                                        /* 1 for remote wardhouse, 0
otherwise */
};

struct ordstat_struct {
    char            invalid; /* transaction completed sucessfully */
    long did;
    long cid;
    char clast[17];
    char byname;                                        /* 1 for by last name, 0 for by id
*/
};

struct delivery_struct {
    char            invalid; /* transaction completed sucessfully */
    char carrier;
};

struct stocklev_struct {
    char            invalid; /* transaction completed sucessfully */
    long threshold;
};

struct generic_struct {
    char            invalid; /* transaction completed sucessfully */
};

union transaction_info {
    char   invalid;
    struct generic_struct  generic;
    struct neword_struct   neword;
    struct payment_struct  payment;
    struct ordstat_struct  ordstat;
    struct delivery_struct delivery;
    struct stocklev_struct stocklev;
};

struct UserGlobal {
    int total_users;
    int max_warehouses;
    int keystroke_sleep;
    int login_timeout;
    int keystroke_packet_size;
    double chances[MAX_TRAN_TYPE];
    double think[MAX_TRAN_TYPE];
    double emulex_response[MAX_TRAN_TYPE];
    double emulex_menu    [MAX_TRAN_TYPE];
};

struct UserLocal {
    int Warehouse;
    int District;
};

struct user_data_header {
};

extern UserGlobal *shmglobal;
extern UserLocal  *shmlocal;

#endif
```

Master Script

```
#define _H_CUR01
#include <cur00.h>
#undef _H_CUR01
extern "C" {
#include "data/cur01.h"
int wrefresh (WINDOW *);
int wclrtoeol(WINDOW *);
int setupterm(char*,FILE*,int*);
int nodelay(int);
int keypad(int);
int wgetch(WINDOW *);
}
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include "data/rte.h"
#include "data/Stats.h"
#include "data/misc.h"
#include "user_tpcc.h"

struct header_s {
    int slave;
    int num;
    int type;
    int num_timestamps;
    int user_data_length;
    int data_type;
};


char *get_variable(char *name);
int get_variable(char *name, int *number);
int send_global_data(void);
int make_ratios (double *buffer);
extern int ramp_up_complete;
extern int interval_start_time, interval_stop_time;
```

```
extern "C" int strcasecmp(char *s1, char *s2);
extern "C" int strncasecmp(char *s1, char *s2, int n);

struct UserSpawnData {
    int Warehouse;
    int District;
};

/* user_master.C */
int user_statistics_print(void);
int user_spawn(int *length, char *buffer);
int user_finished(int length, char *buffer);

extern SlaveStatus slave_status[MAX_SLAVES];

extern Stats status[MAX_TRAN_TYPE][MAX_TIMES];
extern WINDOW *statistics_win;
extern UserGlobal *shmglobal;

/* Transaction mix parameters */
double ratio_desired[6], ratio_min[6], ratio_max[6], ratio_range[6];
char *ratio_names[] = { "RTE", "NEWORDER", "PAYMENT", "ORDSTAT", "DELIVERY",
                        "STOCKLEV", NULL };
char *Status_Names[] = {"Menu", "Keying", "Response", "Think"};


char *transaction_names[] = { "RTE", "New Order", "Payment", "Order Stat",
                          "Delivery", "Stock Level", NULL };

static int current_status = 2, status_needs_refresh = 1;

int user_statistics_print(void) {
    int i;
    static int count = 0;
    double ratios[6];
    if (status_needs_refresh) {
                count = 0;
                status_needs_refresh = 0;
                wmove   (statistics_win, 0, 0);
                wprintw (statistics_win, "%11s %8s %8s %8s %8s %8s %6s %6s %6s",
                                Status_Names[current_status], "90%", "Avg", "Min", "Max",
                                "Samples", "Ratio", "Mix", "Think");
    }
    make_ratios(ratios);


    for (i = 1; i <= 5; i++) {
                /* The reason we do this is because calculating the percentiles
                   is expensive */
                if (count % 10 == 0) {
                    wmove   (statistics_win, i, 0);
                    wprintw (statistics_win, "%11s %8.2f",
                                    transaction_names[i],
status[i][current_status].ninety()/1000.0);
                    count = 0;
                }
                wmove   (statistics_win, i, 21);
                wprintw (statistics_win, "%8.2f %8.2f %8.2f %8d %6.2f %6.2f %6.2f",
                                status[i][current_status].average()/1000.0,
                                status[i][current_status].min()/1000.0,
                                status[i][current_status].max()/1000.0,
                                status[i][current_status].samples(),
                                ratios[i], shmglobal->chances[i],
                                status[i][3].average()/1000.0);
    }
    wmove   (statistics_win, 7, 0);

    extern int runtime_counts[MAX_TRAN_TYPE];
    extern int begin_time, ramp_up, run_time;
    int start = interval_start_time;
    int stop  = interval_stop_time;
    double interval = ((double)(stop-start) / (1000*60));
    double samples = status[1][2].samples();
    if (interval <= 0 || samples <= 0) {
                wprintw (statistics_win, "TPM-C: %7s / ", "-------");
    } else {
                wprintw (statistics_win, "TPM-C: %7.2f / ", samples/interval);
    }
    samples = runtime_counts[1];
    if (samples > 0) {
                start = begin_time+((ramp_up>=0)?ramp_up:0);
                if (run_time > 0 && stop > begin_time + ramp_up + run_time) {
                    stop = begin_time + ramp_up + run_time;
                }
                interval = (double)(stop - start)/(1000.0*60.0);
                wprintw (statistics_win, "%7.2f", samples/interval);
    } else {
                wprintw (statistics_win, "-------");
    }

    count++;
    return RTE_OK;
}

const int MAX_WAREHOUSES=2100;
/* All of this 10 stuff is district size.  Should be a constant.
   Maybe fix that later */
int num_warehouses = -1;
int warehouses[MAX_WAREHOUSES*10];
int user_spawn(int number, int *length, char *buffer) {
    int i, min_index;
    UserSpawnData *ptr = (UserSpawnData *)buffer;
    *length = sizeof(*ptr);

    min_index = 0;
    for (i = 1; i < (num_warehouses)*10 && i < MAX_WAREHOUSES*10; i++) {
                if (warehouses[i] < warehouses[min_index]) {
                    min_index = i;
                }
    }

    ptr->Warehouse = min_index / 10 + 1;
    ptr->District  = min_index % 10 + 1;
    warehouses[min_index]++;
/*     iprint (IPRINT_INFO, "Driver for Warehouse %d, District %d started.  warehouses[%d]++ =
%d\n",
            ptr->Warehouse, ptr->District, min_index, warehouses[min_index]); */
    return RTE_OK;
}

int user_finished(int length, char *buffer) {
    UserSpawnData *ptr = (UserSpawnData *)buffer;
    int temp = (ptr->Warehouse-1)*10+ptr->District-1;
    warehouses[temp]--;
/*     iprint (IPRINT_INFO, "Driver for Warehouse %d, District %d died.  warehouses[%d]-- =
%d\n",
            ptr->Warehouse, ptr->District, temp, warehouses[temp]); */
    return RTE_OK;
}


double limit(double min, double max, double val) {
    if (val < min)
                return min;
    if (val > max)
                return max;
    return val;
}


int make_ratios (double *buffer) {
    int neword   = status[NEWORDER][0].samples();
    int payment  = status[PAYMENT] [0].samples();
    int ordstat  = status[ORDSTAT] [0].samples();
    int delivery = status[DELIVERY][0].samples();
    int stocklev = status[STOCKLEV][0].samples();
    int total    = neword + payment + ordstat + delivery + stocklev;
    int i;

    if (total == 0) {
                buffer[NEWORDER] = 100.0;
                for (i = 2; i < 6; i++) {
                    buffer[i] = ratio_desired[i];
                    buffer[NEWORDER] -= buffer[i];
                }
                return 0;
    }

    buffer[PAYMENT]  = (double)payment  / (double)total * 100.0;
    buffer[ORDSTAT]  = (double)ordstat  / (double)total * 100.0;
    buffer[DELIVERY] = (double)delivery / (double)total * 100.0;
    buffer[STOCKLEV] = (double)stocklev / (double)total * 100.0;
    buffer[NEWORDER] = 100.0- buffer[PAYMENT] - buffer[ORDSTAT] -
                                                buffer[DELIVERY] - buffer[STOCKLEV];

    return total;
}

int user_global_update(int *length, char *buffer) {
    UserGlobal *shmglobal = (UserGlobal *)buffer;
    static double last[6];
    static int users_last=-1;
    double ratios[6];
    double current[6];
    int i, different = 0;
    int desired = 0;

    *length = sizeof(*shmglobal);

    make_ratios(ratios);

    /* Calculate ratios we want for next time */
    /* Note: we just keep on with the desired values until ramp-up is complete
             this at least starts us out without any humps or spikes in the
                    graph */
    if (ramp_up_complete) {
                current[NEWORDER] = 100.0;
                for (i = 2; i < 6; i++) {
                    if (ratio_desired[i] > ratios[i]) {
                                current[i] = ratio_max[i];
                    } else {
                                current[i] = 2*ratio_desired[i] - ratios[i];
                                if (current[i] < ratio_min[i])
                                    current[i] = ratio_min[i];
                    }
                    current[NEWORDER] -= current[i];
                }
    } else {
                for (i = 1; i < 6; i++) {
                    current[i] = ratio_desired[i];
                }
    }

    /* Add up all the users */
    /* This needs to be changed to be more transparent */
    shmglobal->total_users = 0;
    for (i = 0; i < MAX_SLAVES; i++) {
                shmglobal->total_users += slave_status[i].active;
                desired                += slave_status[i].desired;
    }
    /* Count up number of warehouses we WANT to have */
    if (num_warehouses < 0) {
                num_warehouses = (desired-1)/10+1;
    }
    shmglobal->max_warehouses = num_warehouses;


    for (i = 2; i < 6; i++) {
                if (current[i] != last[i])
                    different = 1;
    }

    // Don't send if it's the same as last time
    if ( !different && shmglobal->total_users == users_last ) {
                return RTE_ERROR;
    }

    users_last = shmglobal->total_users;
    for (i = 1; i < 6; i++) {
                shmglobal->chances[i] = last[i] = current[i];
    }

    send_global_data();

    return RTE_OK;
}


int parse_array(char *string, int max, int *buffer) {
    int i, rc;
    char *ptr;
    char *temp = strdup(string);
    ptr = strtok(temp, ",");
    for (i = 0; ptr && i < max; i++) {
                rc = sscanf(ptr, "%d", &buffer[i]);
                if (rc < 1) {
                    free(temp);
                    return i;
                }
                ptr = strtok(NULL, ",");
    }
    free(temp);
    return i;
}

int parse_array(char *string, int max, double *buffer) {
    int i, rc;
    char *ptr;
    char *temp = strdup(string);
    ptr = strtok(temp, ",");
    for (i = 0; ptr && i < max; i++) {
                rc = sscanf(ptr, "%lf", &buffer[i]);
                if (rc < 1) {
                    free(temp);
                    return i;
                }
                ptr = strtok(NULL, ",");
    }
```

```
    free(temp);
    return i;
}


int user_init() {
    double  dbuffer[32];
    int   rc, i;
    char *ptr;

    if (get_variable("KEYSTROKE_SLEEP", &shmglobal->keystroke_sleep) != RTE_OK) {
              shmglobal->keystroke_sleep = 0;
    }
    if (get_variable("LOGIN_TIMEOUT", &shmglobal->login_timeout) != RTE_OK) {
              shmglobal->login_timeout = 120; /* 2 minutes */
    }
    if (get_variable("KEYSTROKE_PACKET_SIZE", &shmglobal->keystroke_packet_size) != RTE_OK) {
              shmglobal->keystroke_packet_size = 0;
    }
    shmglobal->login_timeout *= 1000;
    if (get_variable("WAREHOUSES", &num_warehouses) != RTE_OK) {
              num_warehouses = -1;
    }
    iprint(IPRINT_INFO, "Login Timeout = %s\n", mstoa(shmglobal->login_timeout, 0));
    iprint(IPRINT_INFO, "Keystroke Sleep = %s\n", mstoa(shmglobal->keystroke_sleep*1000, 0));
    iprint(IPRINT_INFO, "Keystroke Packet Size= %d\n", shmglobal->keystroke_packet_size);
    if (num_warehouses >= 0) {
              iprint(IPRINT_INFO, "Fixed Warehouses to = %d\n", num_warehouses);
    }

    if (!(ptr = get_variable("NEWORDER"))) {
              iprint_error ("Error. NEWORDER variable not found\n");
              exit (1);
    }
    if (parse_array(ptr, 3, dbuffer)!=3) {
              iprint_error ("Error. NEWORDER should be think, emulex_menu, emulex_response");
              exit (1);
    }
    shmglobal->think          [NEWORDER] = dbuffer[0];
    shmglobal->emulex_menu    [NEWORDER] = dbuffer[1];
    shmglobal->emulex_response[NEWORDER] = dbuffer[1];

    for (i = 2; i < 6; i++) {
              if (!(ptr = get_variable(ratio_names[i])) ||
                  (parse_array(ptr, 6, dbuffer)!=6)) {
                  iprint(__FILE__, __LINE__, IPRINT_ERROR,
                         "Error. %s should be think, emulex_delay, desired, min, max",
                                ratio_names[i]);
                  exit (1);
              }
              shmglobal->think[i]           = dbuffer[0];
              shmglobal->emulex_menu[i]     = dbuffer[1];
              shmglobal->emulex_response[i] = dbuffer[2];
              ratio_desired[i]              = dbuffer[3];
              ratio_min[i]                  = dbuffer[4];
              ratio_max[i]                  = dbuffer[5];
              ratio_range[i]                = ratio_max[i]-ratio_min[i];
    }

    return RTE_OK;
}

int user_extra_data(header_s *header) {
    int i;
    int num_timestamps;

    if (header->data_type != RTE_ITEM_KEYSTROKE_TIMES)
              return RTE_OK;
    int *times = (int *)((char *)header+sizeof(struct header_s));
    num_timestamps = header->user_data_length / 4 - 1;

    iprint (IPRINT_TRACE, "Keystroke times = ");
    for (i = 0 ; i < num_timestamps; i++) {
              iprint (IPRINT_TRACE, "%d ", times[i]);
    }
    iprint (IPRINT_TRACE, "\n", times[i]);

    return RTE_OK;
}

int user_process_command(char *command) {
    char buffer[256], *ptr;
    int i, found, len;
    strncpy (buffer, command, 256);
    ptr = strtok (buffer, " \t");
    found = 0;
    if (!strcasecmp (ptr, "display")) {
        while (ptr && (ptr = strtok(NULL, " \t"))) {
            if (*ptr == '\0')
                continue;
            for (i = 0; i < 5; i++) {
                len = min(strlen(Status_Names[i]), strlen(ptr));
                if (!strncasecmp (ptr, Status_Names[i], len)) {
                    status_needs_refresh = found = 1;
                    current_status = i;
                    return RTE_OK;
                }
            }
            printf ("Unknown type to display: %s\n", ptr);
        }
    }
    printf ("Unknown Command: %s\n", command);
    return RTE_ERROR;
}


int user_begin() {
    return RTE_OK;
}

void user_make_header(char *buffer) {
    int i;
    struct user_data_header *data = (struct user_data_header *)buffer;
}
```

User Script

```
/*****************************************************************************/
/*                                                                           */
/*  Licensed Materials - Property of IBM                                     */
/*                                                                           */
/*  (c) Copyright IBM Corp. 1994, 1995                                       */
/*                                                                           */
/*  US Government Users Restricted Rights - Use, duplication or              */
/*  disclosure restricted by GSA ADP Schedule Contract with IBM Corp.        */
/*                                                                           */
/*****************************************************************************/
/*****************************************************************************/
/***                    TPCC FILE FOR ALL USERS                            ***/
/*****************************************************************************/
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/time.h>
#include <sys/timers.h>
#include <time.h>
#include "rte_slave.h"
#include "user_tpcc.h"

extern SHM_Slave *shm;
extern TableEntrySlave *shmentry;
extern DriverStatus    *status;
extern echo_trace(char *);
extern echo_trace();
extern char *expect_save;

const char *SQL_TPERRNO_MESSAGE = "tperrno";
const char *SQL_RTN_MESSAGE =     "rtn:";
const char *SQL_FATAL_MESSAGE =   "SQL Fatal Error";
const char *ROLLBACK_MESSAGE =    "ITEM NUMBER IS NOT VALID";

int             WHSEID;                 /* warehouse number for each users */

/*****************************************************************************/
/* The "uniform()" function has range of the absolute value of the        */
/* difference between the min. and the max values upto 2147483647.      */
/*****************************************************************************/
/*-----------------------------------------------------------*/
/* NURand                                */
/*-----------------------------------------------------------*/
/* A: 255 for C_LAST, 1023 for C_ID, 8191 for OL_I_ID   */
/* x: 0 for C_LAST, 1 for C_ID and OL_I_ID                   */
/* y: 999 for C_LAST, 3000 for C_ID, 100000 for OL_I_ID */
/*-----------------------------------------------------------*/
long
NURand(int A, int x, int y, long cval)
{
    return ((((long) uniform((long) 0, (long) A) | (long) uniform((long) x, (long) y)) + cval)
% (y - x + 1)) + x;
}

/*-----------------------------------------------------------*/
/* getname                               */
/*-----------------------------------------------------------*/
/* generates a random number from 0 to 999 inclusive    */
/* a random name is generated by associating a random   */
/* string with each digit of the generated number       */
/* three strings are concatenated to generate lastname  */
/*-----------------------------------------------------------*/
char          *
getname()
{
    char          *last_name_parts[] =
    {
     "BAR",
     "OUGHT",
     "ABLE",
     "PRE",
     "PRES",
     "ESE",
     "ANTI",
     "CALLY",
     "ATION",
     "EING"
    };
    char            lastname[128];
    int             random_num;

    random_num = NURand(255, 0, 999, LASTC);
    strcpy(lastname, last_name_parts[random_num / 100]);
    random_num %= 100;
    strcat(lastname, last_name_parts[random_num / 10]);
    random_num %= 10;
    strcat(lastname, last_name_parts[random_num]);
    return (lastname);
}


#define KEY_PF1 "[001q"
#define KEY_PF2 "[002q"
#define KEY_PF3 "[003q"
#define KEY_PF4 "[004q"
#define KEY_PF5 "[005q"
#define KEY_PF6 "[006q"

/*****************************************************************************/
/***                     Delivery Transaction                              ***/
/*****************************************************************************/
int
Delivery()
{
    static struct delivery_struct delivery, delivery_new;
    char            carrier[32];
    int             resp, charlen;
    double          think_max, think_DEL;
    char           const *ptr;

    delivery_new.invalid = 0;

    /* Calculate what we are going to enter next time */
    delivery_new.carrier = uniform(1, 10);   /* carrier # 1 to 10 */
    sprintf(carrier, "%d\n", delivery_new.carrier);


    set_typing_delay(0);
    transaction_sleep_do();
    transaction_start(DELIVERY, sizeof(delivery), &delivery);              /* logging */
    delivery = delivery_new;
    echo_trace ("Waiting for Menu (DELIVERY)");
    transmit(KEY_PF4);        /* menu option 4 */
    resp = expect("CMD4IIoff");             /* screen position */
    if (resp == ERROR) {
              iprint (IPRINT_ERROR, "Slave %d: Failed to receive Delivery screen\n",
shmentry->num);
              return (ERROR);
    }
    charlen = strlen(carrier);
    set_typing_delay(2000 / charlen + 1);
    usleep(shmglobal->emulex_menu[DELIVERY]*1000000.0+0.9);
    transaction_mark(WHERE_NOW);            /* end of menu, beginning of keying
                                                                          * time */
    echo_trace ("Keying");
    transmit(carrier);
    transaction_mark(WHERE_NOW);            /* end of keying, beginning of
                                                                          * response time */
    echo_trace ("Wait for Response");
    set_typing_delay(0);
    transmit("\r");
//    resp = expect("Delivery has been queued");
```

```
    resp = expect("EnterIIoff");

    if (resp == ERROR) {
            iprint (IPRINT_ERROR, "Slave %d: Failed to receive Delivery response\n",
shmentry->num);
            return (ERROR);
    }
    usleep(shmglobal->emulex_response[DELIVERY]*1000000.0+0.9);
    transaction_mark(WHERE_NOW);            /* end of response, beginning of
                                                                        * think time */
    if (expect_after_match (SQL_TPERRNO_MESSAGE)) {
            delivery.invalid = 1;
            if (ptr = expect_after_match (SQL_RTN_MESSAGE)) {
                iprint (IPRINT_ERROR, "Slave %d: Delivery found '%s'\n",
                        shmentry->num, ptr);
            } else {
                iprint (IPRINT_ERROR, "Slave %d: Delivery found '%s'\n",
                        shmentry->num, SQL_TPERRNO_MESSAGE);
            }
            return RTE_ERROR;
    }
    if (expect_after_match (SQL_FATAL_MESSAGE)) {
            delivery.invalid = 1;
            iprint (IPRINT_ERROR, "Slave %d: Delivery found '%s'\n", shmentry->num,
SQL_FATAL_MESSAGE);
            return RTE_ERROR;
    }
    echo_trace ("Thinking");
    transaction_sleep_set(neg_exp_4(shmglobal->think[DELIVERY])*1000.0);
    return (RTE_OK);
}

/*********************************************************************/
/***                  New Order Transaction                        ***/
/*********************************************************************/
int
NewOrder()
{
    static struct neword_struct neword, neword_new;
    int             RBtrans, remoteflag, i, resp, charlen, whses, low_whse=1;
    double          think_max, think_NO;
    char            buff[1024], district[32], customer[32], warehouse[32], item[32],
quantity[32];
    char            const *ptr;


    neword_new.rollback=0;
    neword_new.invalid = 0;

    /*** SECTION TO DETERMINE ROLLBACK TRANSACTION FOR 1% OF NEW ORDERS ***/
    RBtrans = 0;
    if (uniform(1, 5000) <= 50)
            RBtrans = 1;

    neword_new.did = uniform(1, 10);        /* district number */
    sprintf(district, "%d\n", neword_new.did);
    strcpy(buff, district);
    neword_new.cid = NURand(1023, 1, 3000, CUSTC);  /* customer # 1 to 3000 */
    sprintf(customer, "%d\n", neword_new.cid);
    strcat(buff, customer);

    neword_new.nloop = uniform(5, 15);      /* PAW1 for 5 to 15 iterations */
    remoteflag = 0;                         /* for remote orders */
    neword_new.olremote=0;    /* find total number of remote order-lines */

    whses = shmglobal->max_warehouses;

    for (i = 0; i < neword_new.nloop; i++) {
            neword_new.item[i].olswid = WHSEID;
            if (whses > 1 && (uniform(0.0, 100.0) < 1.0)) {
                /* for 1% of items (if * uniform()==0) */
                /* Generate a uniform whse number that's different from WHSEID */
                while (neword_new.item[i].olswid == WHSEID) {
                        neword_new.item[i].olswid =
                            (long) uniform((long) low_whse, (long)whses);
                }

                neword_new.olremote++;    /* find total number of remote order-lines */
                remoteflag = 1;
            }
            sprintf(warehouse, "%d\n", neword_new.item[i].olswid);
            strcat(buff, warehouse);

            if (neword_new.nloop == i+1 && RBtrans)      /* If last iteration of loop */
                neword_new.item[i].oliid = 999999;       /* and a RollBack
                                                                                *
transaction. */
            else {
                neword_new.item[i].oliid = NURand(8191, 1, 100000, ITEMC);            /*
item 1 to 100,000 */
            }
            sprintf(item, "%d\n", neword_new.item[i].oliid);
            strcat(buff, (item));
            neword_new.item[i].olquantity = uniform(1, 10);            /* quantity 1 to 10
*/
            sprintf(quantity, "%d\n", neword_new.item[i].olquantity);
            strcat(buff, quantity);
    }                                               /* end of for n_loop */

    if (remoteflag == 1)      /* 1 for remote orders */
            neword_new.oremote = 1;
    else
            neword_new.oremote = 0;

    transaction_sleep_do();
    transaction_start(NEWORDER, sizeof(neword), &neword);   /* logging */
    neword = neword_new;
    echo_trace ("Waiting for Menu (NEWORDER)");
    set_typing_delay(0);
    transmit(KEY_PF1);          /* menu option 1 */
    resp = expect("CMD1IIoff");
    if (resp == ERROR) {
            iprint (IPRINT_ERROR, "Slave %d: Failed to receive NewOrder screen\n",
shmentry->num);
            return (ERROR);
    }
    usleep(shmglobal->emulex_menu[NEWORDER]*1000000.0+0.9);
    transaction_mark(WHERE_NOW);            /* end of menu, beginning of keying
                                                                        * time */
    echo_trace ("Keying");
    charlen = strlen(buff);
    set_typing_delay(18000 / charlen + 1);
    if (shmentry->flags & TES_FLAG_KEYSTROKE_TIME) {
            transmit_timed(buff);
    } else {
            transmit(buff);
    }
    transaction_mark(WHERE_NOW);            /* end of keying, beginning of
                                                                        * response time */
    echo_trace ("Wait for Response");
    set_typing_delay(0);
    transmit("\r");
    resp = expect("EnterIIoff");
    if (resp == ERROR) {
            iprint (IPRINT_ERROR, "Slave %d: Failed to receive New Order response\n",
shmentry->num);
            return (ERROR);
    }
    usleep(shmglobal->emulex_response[NEWORDER]*1000000.0+0.9);
    transaction_mark(WHERE_NOW);            /* end of response, beginning of
                                                                        * think time */
    /* Grab order id from return string */
    if (RBtrans)
      neword.rollback=1;
/* This does not work with me5250 PAW1
    if (!(ptr = expect_after_match("\033[50D"))) {
            echo_trace ("Didn't find order-id for neworder");
            iprint (IPRINT_ERROR, "Neworder didn't have Order-ID");
            neword.oid = -1;
    } else {
            neword.oid = atoi(ptr+5);
    }
    if (expect_after_match (ROLLBACK_MESSAGE)) {
            neword.rollback=1;
            echo_trace ("Found rollback!\n");
    }
    if (expect_after_match (SQL_TPERRNO_MESSAGE)) {
            neword.invalid = 1;
            if (ptr = expect_after_match (SQL_RTN_MESSAGE)) {
                iprint (IPRINT_ERROR, "Slave %d: New Order found '%s'\n",
                        shmentry->num, ptr);
            } else {
                iprint (IPRINT_ERROR, "Slave %d: New Order found '%s'\n",
                        shmentry->num, SQL_TPERRNO_MESSAGE);
            }
            return RTE_ERROR;
    }
    if (expect_after_match (SQL_FATAL_MESSAGE)) {
            iprint (IPRINT_ERROR, "Slave %d: New Order found '%s'\n", shmentry->num,
SQL_FATAL_MESSAGE);
            neword.invalid = 1;
            echo_trace ("Fatal Error Message");
            return RTE_ERROR;
    }
*/  // PAW1 end stuff does not work with me5250
    echo_trace ("Thinking");
    transaction_sleep_set(neg_exp_4(shmglobal->think[NEWORDER])*1000.0);

    if (shmentry->flags & TES_FLAG_KEYSTROKE_TIME) {
            log_data(RTE_ITEM_KEYSTROKE_TIMES,
keystroke_length*sizeof(int),keystroke_times);
    }
    return (RTE_OK);
}

/*********************************************************************/
/***                 Order Status Transaction                      ***/
/*********************************************************************/
int
OrderStatus()
{
    static struct ordstat_struct ordstat, ordstat_new;
    char            buff[1024], district[32], customer[32];
    int             resp, charlen;
    double          think_max, think_OS;
    char            const *ptr;

    ordstat_new.invalid = 0;

    ordstat_new.did = uniform(1, 10);       /* district number 1 to 10 */
    sprintf(district, "%d\n", ordstat_new.did);
    strcpy(buff, district);
    if (uniform(1, 100) <= 60) {/* for 60% of transactions */
            strcpy(ordstat_new.clast, getname());       /* by customer last name */
            if (ordstat_new.clast[0] < 'A' || ordstat_new.clast[0] > 'Z') {
                iprint (IPRINT_ERROR, "ASSERTION: OrderStatus getname() returns invalid
name! '%s'\n", ordstat_new.clast);
            }
            sprintf(customer, "\n%s\n", ordstat_new.clast);
            ordstat_new.byname = 1;
            ordstat_new.cid = 0;
    } else {
            ordstat_new.cid = NURand(1023, 1, 3000, CUSTC);             /* cust. # 1 to
3000 */
            sprintf(customer, "%d\n\n", ordstat_new.cid);
            ordstat_new.byname = 0;
            ordstat_new.clast[0] = (char) NULL;
    }
    strcat(buff, customer);

    set_typing_delay(0);
    transaction_sleep_do();
    transaction_start (ORDSTAT, sizeof(ordstat), &ordstat); /* logging */
    ordstat = ordstat_new;
    echo_trace ("Waiting for Menu (ORDSTAT)");
    transmit(KEY_PF5);        /* menu option 3 */
    resp = expect("CMD5IIoff");             /* screen position */
    if (resp == ERROR) {
            iprint (IPRINT_ERROR, "Slave %d: Failed to receive Order Status screen\n",
shmentry->num);
            return (ERROR);
    }
    charlen = strlen(buff);
    set_typing_delay(2000 / charlen + 1);
    usleep(shmglobal->emulex_menu[ORDSTAT]*1000000.0+0.9);
    transaction_mark(WHERE_NOW);            /* end of menu, beginning of keying
                                                                        * time */
    echo_trace ("Keying");
    transmit(buff);

    transaction_mark(WHERE_NOW);            /* end of keying, beginning of
                                                                        * response time */
    echo_trace ("Wait for Response");
    set_typing_delay(0);
    transmit("\r");
    resp = expect("EnterIIoff");            /* screen position */
    if (resp == ERROR) {
            iprint (IPRINT_ERROR, "Slave %d: Failed to receive Order Status response\n",
shmentry->num);
            return (ERROR);
    }
    usleep(shmglobal->emulex_response[ORDSTAT]*1000000.0+0.9);
    transaction_mark(WHERE_NOW);            /* end of response, beginning of
                                                                        * think time */
    if (expect_after_match (SQL_TPERRNO_MESSAGE)) {
            ordstat.invalid = 1;
            if (ptr = expect_after_match (SQL_RTN_MESSAGE)) {
                iprint (IPRINT_ERROR, "Slave %d: Order status found '%s'\n",
                        shmentry->num, ptr);
            } else {
                iprint (IPRINT_ERROR, "Slave %d: Order status found '%s'\n",
                        shmentry->num, SQL_TPERRNO_MESSAGE);
            }
            return RTE_ERROR;
    }
    if (expect_after_match (SQL_FATAL_MESSAGE)) {
            iprint (IPRINT_ERROR, "Slave %d: Order Status found '%s'\n", shmentry->num,
SQL_FATAL_MESSAGE);
```

```
            ordstat.invalid = 1;
            return RTE_ERROR;
    }
    echo_trace ("Thinking");
    transaction_sleep_set(neg_exp_4(shmglobal->think[ORDSTAT])*1000.0);
    return (RTE_OK);
}

/*********************************************************************/
/***                    Payment Transaction                        ***/
/*********************************************************************/
int
Payment()
{
    static struct payment_struct payment, payment_new;
    int             dollars, cents;
    int             resp, charlen, whses, low_whse = 1;
    double          think_max, think_PAY;
    char            buff[1024], district[32], customer[32], waredist[32], amount[32];
    char           const *ptr;

    payment_new.invalid = 0;
    payment_new.did = uniform(1, 10);       /* district number 1 to 10 */
    sprintf(district, "%d\n", payment_new.did);
    strcpy(buff, district);

    if (uniform(1, 100) <= 60) {/* for 60% of transactions */
            strncpy(payment_new.clast, getname(), 17);   /* by customer last name */
            if (payment_new.clast[0] < 'A' || payment_new.clast[0] > 'Z') {
                iprint (IPRINT_ERROR, "ASSERTION: payment_new getname() returns invalid
name! '%s'\n", payment_new.clast);
                exit (10);
            }
            sprintf(customer, "%s\n", payment_new.clast);
            payment_new.byname = 1;
            payment_new.cid = 0;
    } else {
            payment_new.cid = NURand(1023, 1, 3000, CUSTC);             /* cust. # 1 to
3000 */
            sprintf(customer, "%d\n", payment_new.cid);
            payment_new.byname = 0;
            payment_new.clast[0] = (char) NULL;
    }
    if (payment_new.byname)   /* using C_LAST */
            strcat(buff, "\n");
    else                                /* using C_ID */
            strcat(buff, customer);

    whses = shmglobal->max_warehouses;

    if (whses < 2 || uniform(1, 100) <= 85) {/* for 85 % of transactions */
            payment_new.cwid = WHSEID;
            payment_new.cdid = payment_new.did;
            payment_new.remote = 0;
            /*
             * Pwarehousesame++;                total number of home
             * transactions
             */
    } else {                            /* for 15 % of transactions */
            payment_new.cwid = WHSEID;
            while (payment_new.cwid == WHSEID) {
                payment_new.cwid = (long) uniform((long)low_whse, (long) whses);
                /* warehouse 1 to max whses */
            }

            payment_new.remote = 1;
            payment_new.cdid = uniform(1, 10);          /* district 1 to 10 */
            /*
             * Pwarehousediff++;                total number of remote
             * transactions
             */
    }
    sprintf(waredist, "%d\n%d\n", payment_new.cwid, payment_new.cdid);
    strcat(buff, waredist);

    if (payment_new.byname)   /* using C_LAST */
            strcat(buff, customer);
    else                                /* using C_ID */
            strcat(buff, "\n");

    dollars = uniform(1, 5000);             /* dollar amt = 1 to 5000 */
    if (dollars == 5000)
            cents = 0;
    else
            cents = uniform(0, 99);
    payment_new.amount = ((double) dollars) + ((double) cents) / 100.0;
    sprintf(amount, "%d.%2.2d\n", dollars, cents);
    strcat(buff, amount);     /* 1.00 to 5000.00 */

    set_typing_delay(0);
    transaction_sleep_do();
    transaction_start(PAYMENT, sizeof(payment), &payment); /* logging */
    payment = payment_new;
    echo_trace ("Waiting for Menu (PAYMENT)");
    transmit(KEY_PF2);        /* menu option 2 */
    resp = expect("CMD2IIoff");             /* screen position */
    if (resp == ERROR) {
            iprint (IPRINT_ERROR, "Slave %d: Failed to receive Payment screen\n",
shmentry->num);
            return (ERROR);
    }
    charlen = strlen(buff);
    set_typing_delay(3000 / charlen + 1);
    usleep(shmglobal->emulex_menu[PAYMENT]*1000000.0+0.9);
    transaction_mark(WHERE_NOW);            /* end of menu, beginning of keying
                                                                    * time */
    echo_trace ("Keying");
    transmit(buff);

    transaction_mark(WHERE_NOW);            /* end of keying, beginning of
                                                                * response time */
    echo_trace ("Wait for Response");
    set_typing_delay(0);
    transmit("\r");
    resp = expect("EnterIIoff");                            /* screen position */
    if (resp == ERROR) {
            iprint (IPRINT_ERROR, "Slave %d: Failed to receive Payment response\n",
shmentry->num);
            return (ERROR);
    }
    usleep(shmglobal->emulex_response[PAYMENT]*1000000.0+0.9);
    transaction_mark(WHERE_NOW);            /* end of response, beginning of
                                                                * think time */
    if (expect_after_match (SQL_TPERRNO_MESSAGE)) {
            payment.invalid = 1;
            if (ptr = expect_after_match (SQL_RTN_MESSAGE)) {
                iprint (IPRINT_ERROR, "Slave %d: Payment status found '%s'\n",
                        shmentry->num, ptr);
            } else {
                iprint (IPRINT_ERROR, "Slave %d: Payment status found '%s'\n",
                        shmentry->num, SQL_TPERRNO_MESSAGE);
            }
            return RTE_ERROR;
    }
    if (expect_after_match (SQL_FATAL_MESSAGE)) {
                iprint (IPRINT_ERROR, "Slave %d: Payment found '%s'\n", shmentry->num,
SQL_FATAL_MESSAGE);
            payment.invalid = 1;
            return RTE_ERROR;
    }
    echo_trace ("Thinking");
    transaction_sleep_set(neg_exp_4(shmglobal->think[PAYMENT])*1000.0);
    return (RTE_OK);
}

/*********************************************************************/
/***                  Stock Level Transaction                      ***/
/*********************************************************************/
int
StockLevel()
{
    static struct stocklev_struct stocklevel, stocklevel_new;
    char            threshold[32];
    int             resp, charlen;
    double          think_max, think_SL;
    char           const *ptr;

    stocklevel_new.invalid = 0;
    stocklevel_new.threshold = uniform(10, 20);             /* uniform no. between 10 and
                                                                                    *
20 */
    sprintf(threshold, "%d\n", stocklevel_new.threshold);

    set_typing_delay(0);
    transaction_sleep_do();
    transaction_start(STOCKLEV, sizeof(stocklevel), &stocklevel); /* logging */
    stocklevel = stocklevel_new;
    echo_trace ("Waiting for Menu (STOCKLEV)");
    transmit(KEY_PF6);        /* menu option 5 */
    resp = expect("CMD6IIoff");  /* screen position */
    if (resp == ERROR) {
            iprint (IPRINT_ERROR, "Slave %d: Failed to receive Stock Level screen\n",
shmentry->num);
            return (ERROR);
    }
    charlen = strlen(threshold);
    set_typing_delay(2000 / charlen + 1);
    usleep(shmglobal->emulex_menu[STOCKLEV]*1000000.0+0.9);
    transaction_mark(WHERE_NOW);            /* end of menu, beginning of keying
                                                                    * time */
    echo_trace ("Keying");
    transmit(threshold);

    transaction_mark(WHERE_NOW);            /* end of keying, beginning of
                                                                * response time */
    echo_trace ("Wait for Response");
    set_typing_delay(0);
    transmit("\r");
    resp = expect("EnterIIoff");            /* screen position */
    if (resp == ERROR) {
            iprint (IPRINT_ERROR, "Slave %d: Failed to receive Stock Level response\n",
shmentry->num);
            return (ERROR);
    }
    usleep(shmglobal->emulex_response[STOCKLEV]*1000000.0+0.9);
    transaction_mark(WHERE_NOW);            /* end of response, beginning of
                                                                * think time */
    if (expect_after_match (SQL_TPERRNO_MESSAGE)) {
            stocklevel.invalid = 1;
            if (ptr = expect_after_match (SQL_RTN_MESSAGE)) {
                iprint (IPRINT_ERROR, "Slave %d: Stock Level status found '%s'\n",
                        shmentry->num, ptr);
            } else {
                iprint (IPRINT_ERROR, "Slave %d: Stock Level status found '%s'\n",
                        shmentry->num, SQL_TPERRNO_MESSAGE);
            }
            return RTE_ERROR;
    }
    if (expect_after_match (SQL_FATAL_MESSAGE)) {
            iprint (IPRINT_ERROR, "Slave %d: Stock Level found '%s'\n", shmentry->num,
SQL_FATAL_MESSAGE);
            stocklevel.invalid = 1;
            return RTE_ERROR;
    }
    echo_trace ("Thinking");
    transaction_sleep_set(neg_exp_4(shmglobal->think[STOCKLEV])*1000.0);
    return (RTE_OK);
}

/*********************************************************************/
/***  MAIN()                                                       ***/
/*********************************************************************/
int
user_transaction()
{
    char            logout[32];
    double          ntask;
    int             resp;

    if (shmentry->flags & TES_FLAG_KEYSTROKE_TIME) {
            int rc;
            /* Wait for specified period of time */
            sleep (shmglobal->keystroke_sleep);
            /* Quit after one transaction */
            shm->lock(shmentry->pid);
            shmentry->flags |= TES_FLAG_DIE;
            shm->unlock(shmentry->pid);
            rc = NewOrder();
            iprint (IPRINT_INFO, "Slave %d: Keystroke timing setting die flag\n",
shmentry->num);
            return rc;
    }
    /*******************************************************************/
    /***  CHOOSE ONE OF THE TRANSACTIONS                             ***/
    /*******************************************************************/
    ntask = (double) uniform(0.0, 100.0);
    if (ntask <= shmglobal->chances[DELIVERY])
            return Delivery();
    ntask -= shmglobal->chances[DELIVERY];
    if (ntask <= shmglobal->chances[ORDSTAT])
            return OrderStatus();
    ntask -= shmglobal->chances[ORDSTAT];
    if (ntask <= shmglobal->chances[PAYMENT])
            return Payment();
    ntask -= shmglobal->chances[PAYMENT];
    if (ntask <= shmglobal->chances[STOCKLEV])
            return StockLevel();
    return NewOrder();

#if 0
    if (resp != RTE_OK) {                   /* logoff if response is not correct */
            strcpy(logout, "9\r");          /* menu option 9 */
            transmit(logout);
            resp = expect("tpcc_cstux_inf:");
            return (ERROR);
    } else
            return (RTE_OK);
#endif
```

```
}                                                       /* end of main */

int user_parameter_change(void) {
    #if 0
    int i;
    iprint(IPRINT_TRACE, "Slave %d: total_users = %d\n", shmentry->num);
    iprint(IPRINT_TRACE, "Slave %d: chances     = ", shmentry->num);
    for (i = 0; i < MAX_TRAN_TYPE; i++)
              iprint(IPRINT_TRACE, "%6.2f ", shmglobal->chances[i]);
    iprint(IPRINT_TRACE, "\nSlave %d: think       = ", shmentry->num);
    for (i = 0; i < MAX_TRAN_TYPE; i++)
              iprint(IPRINT_TRACE, "%6.2f ", shmglobal->think[i]);
    iprint(IPRINT_TRACE, "\n");
    #endif
    return RTE_OK;
}

const int MAX_LOGIN_RETRY = 500;
int user_login(char *user, char *password, void *data) {
    extern char ttyname_original[TTYNAME_LEN];
    UserLocal *localdata = (UserLocal *)data;
    int rc;
    int retry = 0;
    int timeout_value = shmglobal->login_timeout;
    char buffer[256];
              char outBuf[8192];
              int i;
              char test;

    set_typing_delay(0);

    iprint(IPRINT_INFO, "slave %d: looking for login prompt!!!\n",
        shmentry->num);

    rc = expect("IBM", timeout_value);
    if (rc != RTE_OK) {
        iprint(IPRINT_ERROR, "slave %d: didn't get login prompt.\n");
        return RTE_ERROR;
    }

    rc = expect("(B", timeout_value);
    if (rc != RTE_OK) {
        iprint(IPRINT_ERROR, "slave %d: didn't get login prompt.\n");
        return RTE_ERROR;
    }

    iprint(IPRINT_INFO, "slave %d: got login prompt!!!!.\n", shmentry->num);
    rc = transmit (user);
    rc = transmit ("\t");
    rc = transmit (password);
    rc = transmit ("\r");
    iprint(IPRINT_INFO,
        "slave %d: sent username/password; waiting for response.\n",
        shmentry->num);

    rc = expect2 ("TPCCUSER","\033[18;7H"); // JDH

    if (rc == RTE_ERROR) {
        iprint (IPRINT_ERROR, "Slave %d: Failed expecting prompt\n",
            shmentry->num);
              return RTE_ERROR;
    }
    iprint(IPRINT_INFO,
        "slave %d: signed on; return code %d\n", shmentry->num, rc);
    if (rc == 1)  // Handle message about user already signed on
    {
       iprint(IPRINT_INFO, "Slave %d:Handling message about user signed on\n");
       rc = transmit ("\r");
       expect("\033[18;7H");
    }
    iprint(IPRINT_INFO, "slave %d: got response; starting application\n",
        shmentry->num);

    /* PAW1 Add code to set the time on the AS/400 to match the time */
    /* on the RS/6000s. Then make a call to synch the clocks of all  */
    /* of the AS/400s. Only the first user to sign on will do this.  */
#ifdef SYNC_TIMES
    if (localdata->Warehouse == 1 && localdata->District == 1)
    {
              time_t Tp;
        char * timeString = new char[26];
        strcpy (buffer, "chgsysval QTIME value('");
        time(&Tp);
        timeString = ctime(&Tp);
        strncat(buffer, &timeString[11], 2);
        strncat(buffer, &timeString[14], 2);
        strncat(buffer, &timeString[17], 2);
        strcat(buffer,"')\r");
        transmit(buffer);
        expect("[18;7H");
        delete timeString;
        transmit("call SYNCTIMES\r");
        expect("[18;7H");

    }
#endif

// FILE *fp=fopen("chkwd", "w");                //     VKK

//       fprintf(fp, "fprintf testing\n");  fflush(fp);

    iprint(IPRINT_INFO, "about to run firstpgm1\n");

              rc = transmit("firstpgm1\r");
    rc = expect("\033[19C");

    iprint(IPRINT_INFO, "slave %d: entering home warehouse/district.\n",

        shmentry->num);

   // sprintf(buffer, "%04d%02d\r",
   //         localdata->Warehouse, localdata->District);
   // rc = transmit (buffer);
   // rc = expect ("\033[5;37H", timeout_value);
              iprint(IPRINT_INFO, "slave %d, about to do expect", shmentry->num);
              rc = expect("Warehouse", timeout_value);
              iprint(IPRINT_INFO, "expect return code = %d", rc);

//            rc = expect("[[C^[[", timeout_value);
for (i = 0; i < 35; i++)
{
              //iprint(IPRINT_INFO, "%d is %c\n", i, readChar());
              incIndex();
}
for (i = 0; i < 4; i++)
{
              buffer[i] = readChar();
              incIndex();
}
buffer[4] = '\0';
iprint(IPRINT_INFO, "slave: %d, warehouse is: %s\n", shmentry->num,
                                        buffer);
localdata->Warehouse = atoi(buffer);
//sprintf(buffer, "%04d", localdata->Warehouse);
iprint(IPRINT_INFO, "vinay test wrhs %d vinays test wrhs\n", localdata->Warehouse);
// fprintf(fp, "Warehouse id is %d\n", localdata->Warehouse); fflush(fp);  //     VKK

              rc = expect("District", timeout_value);
              //incIndex(); incIndex();
              //buffer[0] = readChar();
              //incIndex();
              //buffer[1] = readChar();
              //buffer[2] = '\0';
              for (i = 0; i < 41; i++)
              {
                        //iprint(IPRINT_INFO, "dist; %d is %c\n", i, readChar());
                        incIndex();
              }

              buffer[0] = readChar();
              incIndex();
              buffer[1] = readChar();
              buffer[2] = '\0';

              localdata->District = atoi(buffer);
              iprint(IPRINT_INFO, "slave: %d, district is vinays test dst %d vinays test
dst\n",
                                                  shmentry->num, localdata->District);
// fprintf(fp, "District id is %d\n", localdata->District);  fflush(fp); // VKK

              sprintf(buffer, "\r");
              rc=transmit(buffer);
              // Send it again
//    sprintf(buffer, "%04d%02d\r",
  //          localdata->Warehouse, localdata->District);
    //iprint(IPRINT_INFO, "slave %d buffer = %s", shmentry->num, buffer);
              sprintf(buffer, "\r");
              rc = transmit (buffer);
    rc = expect ("\033[2;29H", timeout_value);


    if (rc != RTE_OK) {
        iprint (IPRINT_ERROR, "Slave %d: Failed expecting initial screen\n",
              shmentry->num);
              return RTE_ERROR;
    }
    iprint(IPRINT_INFO, "slave %d: logged in.\n", shmentry->num);
    /*PAW1 Now switch the emulator so it does not update the screen */
    transmit("\004");  /* CTRL-D will switch to no show mode */
    transmit("J");  /* Tells 5250 to keep going, not quit */


    WHSEID = localdata->Warehouse;
//            fprintf(fp, "WHSEID = %d\n", WHSEID); fflush(fp);     //    VKK

//            fclose(fp);                //         VKK
              return RTE_OK;
}

int user_init () {
//    FILE *fp1=fopen("chkwrhs","w");            // VKK
//     fprintf(fp1, "checking wrhs\n"); fflush(fp1);       //   VKK
    extern int expect_save_active;
    WHSEID = shmlocal->Warehouse;
//  fprintf(fp1,"wrhs is %d\n", WHSEID);    fflush(fp1);        //       VKK
    status->max_transmit =  shmglobal->keystroke_packet_size;
    expect_save_active = 1;
    return RTE_OK;
//            fclose (fp1);        //       VKK
}

int user_cleanup () {
    transmit(KEY_PF3);        /* PAW1 exit tpcc application */
    expect("IIoff");
    transmit("signoff\r");    /* end user's session */
    expect("IIoff");
    transmit("\004\004");  /* quits out of e5250 */
    transaction_start(0, 0, NULL); // Just something to clear out the buffer...
    return RTE_OK;
}
```

Appendix F. 180-Day DASD Requirements

The appendix documents the information required to calculate the 180-Day DASD requirements for the TPC Benchmark C measurements on the IBM AS/400e server 40S-2208 system. Also included is the calculation used to determine the amount of DASD required for the 8 hours of journal space.

Section "IBM AS/400e server 40S-2208 -- 180-Day DASD Requirements" shows the values in the calculation of hte 180-Day Space. Also used were the formulas provided in Clause 4.2.3 of the TPC Benchmark C Standard Specification. These formulas are:

Daily-Growth	=	{dynamic-space/(W*62.5)} *tpmC
Daily-Spread	=	MAX (0, Free-Space - 1.5*Daily-Growth)
180-Day-Space	=	Static-Space + 180*(Daily-Growth+Daily-Spread)

IBM AS/400 server 40S-2208--180-Day DASD Requirements

Static Files	Number of Warehouses 3700			tpmC	43169.85	
File name	Customer	District	Item	New Order	Stock	Warehouse
Number of Records	111000000	37000	100000	33300000	370000000	3700
Data Space	74149081088	4202496	9445376	366329856	113591304192	401408
Index Space	1977901056	794624	1843200	607674368	5951741952	106496
Add'l Index	6273720320		8192	12288	12288	
Add'l Index	12288					
Dynamic Files						
File name	History	Orderline	Orders			
Initial Records	111000000	1109993862	111000000			
Size Per Record	49.0017825826	55.0018629189	**********			
Initial Data Space	5439197867	61051730239	2886109867			
Initial Index Space		28509769728	3645005824			
Add'l Index		12288	1830838272			
Add'l Index						
Configured Space	6527037440	73262481408	3463331840			
Free Space	1087839573	12210751169	577221973			
Static Space	247066947174					
Dynamic Space	69377037972					
Free Space	13875812716					
Daily Growth	12951335449					
Daily Spread	0					
180 Day Space	2578307328080					
System Software	1941358080					
SubTotal	2580248686160					
Journal Space	295267253456					
Total DASD Required	2875515939616					
DASD Priced	3065084000000					
Difference	189568060384					

IBM AS/400 server 40S-2208--Journal DASD Requirements

To determine the amount of DASD required for the 8 hours of journal, a series of tests were run for an extended period. Through these tests it was determined that an average of 6,839,664 bytes of DASD are required per tpmC.

Appendix G. Auditor Letter

Information Paradigm

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL

Certified Auditor

Test Sponsor: Karl R. Huppler
Advisory Programmer
IBM Corp Dept 53G
3605 Highway 52 N.
Rochester, MN 55901

August 27, 1998

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

Platform: AS/400 9406 Model S40 with FC 2208
DataBase Manager: DB2/400 Integrated Relational Database V4 R3
Operating System: OS/400 Version 4 Release 3
Transaction Manager: CICS for OS/400 V4 R3

The results were:

CPUs	Memory	Disks	NewOrder 90% Response Time	tpmC
Server: AS/400 9406 Model S40 with FC 2208				
12 x 262 MHz PowerPC (9406 S40-2208)	40 GB	316 x 8 GB 40 x 17 GB	.71 Seconds	43,169.85
Twenty Clients (see note), specification for each AS/400 9401-0294 priced client				
1 x 50 MHz PowerPC (9401 150-0294)	192 MB	2 x 4 GB	n/a	n/a

In my opinion, these performance results were produced in compliance with the TPC 3.3 requirements for the benchmark. The following verification items were given special attention:

- The transactions were correctly implemented
- The database records were the proper size
- The database was properly scaled and populated

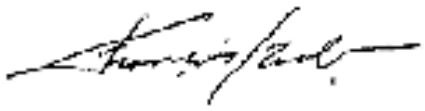
1373 North Franklin Street • Colorado Springs, CO 80903-2627 • Office: 719-473-7555 • Fax: 719-473-7556

- The ACID properties were met
- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured.
- At least 90% of all delivery transactions met the 80 Second completion time limit
- All 90% response times were under the specified maximums
- The measurement interval was representative of steady state conditions
- The reported measurement interval was 20 minutes
- At least 5 file synchronization cycles occurred during the measurement interval
- Measurement repeatability was verified
- The 180 day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

The tested configuration included (11) priced clients model AS/400 9401-150 (192 MB) and (9) non-priced clients with model numbers as follows: (6) AS/400 9406-170 (3.5 GB), (1) AS/400 9406-830 (16 GB), (1) AS/400 9406-170 (4 GB) and (1) AS/400 9406-520 (4.5 GB). The priced configuration includes (97) AS/400 9401-150 systems (192 MB). Based on data collected on all the individual priced and non priced clients, it is my opinion that this substitution would have no negative affect on the reported performance.

Respectfully Yours,



François Raab
President

AS/400 9406 Model S40

1978 North Franklin Street · Colorado Springs, CO 80907-2527 · Office: 719/473-7555 · Fax: 719/473-7554