Burroughs

Installation Manual

B 1000 Series On-Line Data Entry System (ODESY)

Priced Item
Printed in U.S.A.
May 1983

1129384

Burroughs

Installation Manual

B 1000 Series On-Line Data Entry System (ODESY)

Priced Item
Printed in U.S.A.
May 1983

1129384

The names, places and/or events depicted herein are not intended to correspond to any individual, group or association existing, living or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual living or otherwise, or that of any group or association is purely coincidental and unintentional.

The within information is not intended to be nor should such be construed as an affirmation of fact, representation or warranty by Burroughs Corporation of any type, kind or character. Any product and related materials disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license agreement. The only warranties made by Burroughs with respect to the products described in this material are set forth in the above mentioned agreement.

The Customer should exercise care to assure that use of the materials will be in full compliance with laws, rules and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change. Revisions may be issued from time to time to advise of changes and/or additions.

Any comments or suggestions regarding this publication should be forwarded to Documentation, Systems Development Group, Burroughs Corporation, 3519 West Warner Avenue, Santa Ana, CA 92704.

TABLE OF CONTENTS

INTRODUCTION

This manual describes how to install and maintain the Burroughs B 1000 Series On-Line Data Entry System (ODESY). It provides an overview of the installation process by approaching that process as a series of phases. Additionally, this manual supplies the manager and system operator with information which is critical to the effective operation of ODESY.

ODESY is a sophisticated data entry and verification system which uses multiple on-line terminal display units. ODESY enables you to enter, retrieve, and correct information at the terminal. This speeds up the data entry process and facilitates the processing of error-free data. This also eliminates the complication and inconvenience of a separate data preparation department.

The primary features of ODESY are:

1. On-line data entry, verification, and correction.
2. Application independence.
3. Data entry at source.
4. Inquiry facilities.
5. Operator and data security.

The manual is divided into 10 sections and six appendices. Section 1 provides an overview of the installation process, and lists hardware and software requirements.

Section 2 explains in detail the specifics of installing ODESY.

Section 3 describes the various master files associated with ODESY.

System parameters, the data used to control and restrict general system operations, are contained in Section 4.

Section 5 presents the content and design of the screen formats used by terminal operators to enter data.

An illustrated example is included in Section 6 with a detailed explanation of the program used to create screen formats.

Section 7 highlights the programs used by terminal operators. These programs and the data entry procedure are fully documented in the B 1000 Series On-Line Data Entry System (ODESY) Terminal Operator's Manual, form 1131851.

Section 8 outlines the auxiliary programs which are basically management, conversion, and debugging aids to system operation.

The inquiry facilities available with ODESY and the operation of user programs, including two worked examples, are covered in Section 9.

Section 10 discusses various topics that relate to system maintenance, such as the control of data, security, and the backup and recovery of files.

Appendix A lists the commands used with the Format Maintenance Program, and explains how and when to use each one.

Appendix B contains the input formats used with the File Maintenance and Format Maintenance Programs, and a brief definition of the content of each screen.

A detailed discussion of check digit verification schemes is provided in Appendix C.

Appendix D lists the error and information messages associated with the Format Maintenance Program, and includes an explanation of each.

Appendix E contains system error documentation for the Extract program.

Appendix F lists the files for all of the ODESY programs.

The style identification of ODESY is B1000 DE2.

REFERENCE MATERIALS

This manual is designed for use with these manuals:

1. B 1000 Series On-Line Data Entry System (ODESY) Capabilities Manual, form 1131893.
2. B 1000 Series On-Line Data Entry System (ODESY) Terminal Operator's Manual, form 1131851.
3. B 1700/B 1800 Systems Network Definition Language (NDL) Reference Manual, form 1073715.
4. B 1000 Systems System Software Operation Guide, form 1108982.
5. TD 700/TD 800 Equipment Reference Manual, form 1068699.
6. TD 820 Equipment Reference Manual, form 1085685.
7. TD 730/TD 830 Systems Reference Manual, form 1093788.
8. B 1700/B 1800 Systems Software Release Mark X.0 Document/SMCS.
9. B 1000 Generalized Message Control System (GEMCOS) User's Manual, form 1093499.

SECTION 1

INSTALLATION OVERVIEW

ODESY is a sophisticated on-line data entry system. Various system features help to expedite the installation process.

INSTALLATION PHASES

For your convenience the installation process can be broken down into two phases: tasks performed in the days and weeks prior to the actual installation date (pre-installation) and tasks performed as or immediately before the system is brought into "live" operation (installation). Section 2 provides a suggested list of procedures to perform in each phase of installation.

PRE-INSTALLATION TASKS

Pre-installation is the time during which you should study the system thoroughly and develop a comprehensive installation plan. Complete the training of key personnel and begin training all other personnel affected by the new system. You should complete the ordering of all required hardware, forms, and supplies, configure each terminal to be used with ODESY, and evaluate the impact of the system on existing operating procedures and job responsibilities.

Before operating ODESY, you should determine the basic capabilities and options, and gain familiarity with your hardware as it relates to the operation of this system. Install the data communications software. Determine the information you will need to create formats and design several formats according to the requirements of your business.

INSTALLATION TASKS

All of the previous activities should be completed by the date of installation. Once ODESY is operating, it can be reconfigured easily. Therefore, the primary aim at the time of installation is to get one data entry station operating, even if only temporarily, and then to follow the

reconfiguration procedures. On implementation day, load the release software onto the system disk. Make any necessary modifications to the standard system configuration distributed with ODESY. Enter and test the formats you designed previously, and begin data entry.

HARDWARE AND SOFTWARE REQUIREMENTS

Listed below are the hardware and software requirements for the installation of ODESY. Study these requirements carefully, and then assess your own hardware and software needs; some of these requirements may necessitate adding to or adjusting your equipment.

HARDWARE REQUIREMENTS

The following is the minimum hardware configuration required:

1. One B 1000 Series system mainframe with data communications and at least 42 kilobytes of memory excluding the Master Control Program. This does not include the Network Controller.

2. At least one terminal display unit. These terminal devices are acceptable: TD 700, TD 730, TD 800, TD 820, and TD 830. A terminal which functions as one of these devices is acceptable; a MT 983, for example, may be used instead of a TD 830.

3. One 4.6 megabyte disk drive.

4. One line printer.

SOFTWARE REQUIREMENTS

The following software is required:

1. Master Control Program (MCP), level 10.0 or greater, and related system software.

2. B 1000 Series Network Definition Language (NDL), modified to your specifications if necessary. A sample NDL is included with the release.

SECTION 2

INSTALLATION PROCEDURES

This section describes the specific procedures to follow to install ODESY. Installation is divided into two phases: pre-installation tasks and installation tasks.

PRE-INSTALLATION TASKS

The pre-installation tasks are activities which should be completed prior to the date of system implementation.

1. Study the ODESY manuals thoroughly.

2. Attend a demonstration of ODESY. A Burroughs representative can arrange a visit to a site already using the system or can give a personal demonstration of the data entry and format maintenance facilities on any computer with the necessary hardware.

3. Develop an installation plan.

4. Analyze how the new system will affect current operating procedures and job responsibilities. The exact procedural changes, modifications to existing forms, and changes in job responsibilities should be determined and documented.

5. Begin and complete the training of key personnel who will control operation of the programs, monitor data passing through the system, and design and maintain formats.

6. Order and install the necessary hardware.

7. Determine the placement of terminals and cables from the terminals to the computer.

8. Configure each terminal display unit. These field engineer settings are required for the TD 730, TD 820, and TD 830:

    1. Programmatic mode control enabled.
    2. Field overflow inhibit disabled.

3. Tabulation field identifier disabled.

These settings are recommended:

1. Carriage return symbol display disabled.
2. End-of-text symbol display disabled.

9. Design formats according to Section 5. Initially, to become familiar with the features of ODESY and minimize the effects of errors, choose just one or two applications. Also, it is best to avoid user programs at first; the user program interface is more easily understood once ODESY is operating and familiar.

10. Install the data communications software. This software forms the interface between the ODESY programs and the stations in the network. Refer to the Data communications information later in this section.

INSTALLATION TASKS

These tasks should be performed in the indicated order at the time of installation. These procedures are written for new users; if you are converting from a previous release, refer to the conversion procedures in the release letter.

1. Copy the software onto a system disk and save the release media for backup purposes. If the ODESYDATA files are loaded onto a user pack instead, you may need to modify the ODESY programs so they can access the data files. See Appendix F for a list of ODESYDATA file names for each program.

2. Get one data entry station operating. See "Executing ODESY" in Section 2 for more information.

3. Execute the Edit Program. Make any necessary changes to the standard system configuration batch 999. Schedule and extract batch 999. Refer to Reconfiguration expanded later in this section.

4. Execute the File Maintenance Program to make the required changes to the Formats, System, and Tank files.

5. Execute the Format Maintenance Program and enter your formats according to Section 6.

6. Test the formats using the Edit Program and the Extraction Program. Application files can be listed and then executed through the application systems. It is important to detect mistakes in record layouts as early as possible.

7. Train all other personnel who are affected by the system.

8. Begin data entry. It is advisable to operate in parallel with existing data entry methods for a short time. Parallel operation normally continues until the formats are stable and produce consistent data. Monitor the results and change the formats to meet the needs of the data entry operators and the application systems.

9. Discontinue parallel operation when the formats have been fully tested.

10. Evaluate results and decide which other applications are to be transferred to ODESY.

DATA COMMUNICATIONS

A B 1000 Data communications network is composed of three parts:

1. Network Controller (NC).
2. Message Control System (MCS).
3. Application programs which interface with the MCS.

NETWORK CONTROLLER

The Network Controller, responsible for polling and selecting the stations of the ODESY network, is the heart of the data communications system. The function of the Network Controller is to process and supervise the flow of messages between application programs and the remote network (stations). Before operating ODESY, you must generate a Network Controller using the Network Definition Language. The NDL statements define the stations of the network and the communications protocols.

MESSAGE CONTROL SYSTEM

A Message Control System (MCS) is responsible for routing messages between stations and programs. The MCS is more powerful than a normal application program and operates closely with the Network Controller as well as with the Master Control Program. Both EDIT and FORMAINT are simple message control systems and are able to handle ODESY when the stations are dedicated to ODESY use only. In this case, the user-written inquiry programs would be thought of as application programs.

To configure your network so stations are dedicated to ODESY use only, declare the remote file STATIONS in your NDL to contain all the terminals dedicated to ODESY's use. When ODESY is executed, it will attach to all the station in its remote file. These stations can only run ODESY.

However, if you require a more dynamic system in which a station may be switched between several on-line programs, or if you wish to control certain system functions such as file control and security, the Generalized Message Control System (GEMCOS) or Supervisory Message Control System (SMCS) can be used to generate a more comprehensive MCS. In this case, GEMCOS or SMCS would be thought of as the primary or supervisory MCS, and the ODESY programs EDIT and FORMAINT would be called secondary message control systems. The inquiry programs would be application programs. All information entered from a remote station would go through GEMCOS or SMCS before being passed to EDIT or FORMAINT.

In this manual, "SMCS" refers to the product called SMCS, which is itself a supervisory MCS. The term "supervisory MCS" refers to a primary or supervisory MCS, such as GEMCOS, SMCS, or any other MCS which is supervisory.

Generalized Message Control System

B 1000 GEMCOS may be used to generate a suitable supervisory MCS for ODESY. Complete documentation on B 1000 GEMCOS should be supplied with the software.

Declare remote files in the network as follows:

1. The remote file for the supervisory MCS should contain all of the stations that are to be dynamically allocated at the time of execution.

2. The ODESY remote file should be declared as a dummy file.

In the following example, the supervisory MCS file SUPERQUEUE contains the three stations of the network, and the ODESY file STATIONS and the CANDE file CANDEQUEUE are both dummy files. This network allows any station to either remain under GEMCOS or to dynamically attach itself to ODESY or CANDE. These are sample NDL statements for a small GEMCOS/ODESY/CANDE network:

```
DECLARATION:
     NIF = "SUPER"/"NIF".
     MAX TALLY = [1].
     MAX FILES = 20.
$LIBRARY POLLTCTD
$LIBRARY SELECTCTD
$LIBRARY AUTOPOLCTL
TERMINAL TD802:
     REQUEST = POLLTCTD : RECEIVE, SELECTCTD : TRANSMIT.
     BUFFERSIZE = 2000.
     ADDRESS = 2.
     TRANSMISSION = 0.
     TYPE = 42.
TERMINAL TD832:
     REQUEST = POLLTCTD : RECEIVE, SELECTCTD : TRANSMIT.
     BUFFERSIZE = 2000.
     ADDRESS = 2.
     TRANSMISSION = 0.
     TYPE = 46.
STATION DEFAULT TD8:
     MYUSE = INPUT,OUTPUT.
     RETRY = 5.
     CONTROLLER = FALSE.
     TERMINAL = TD802.
STATION TD800A:
     DEFAULT = TD8.
     ADDRESS = "1B".
STATION TD800B:
     DEFAULT = TD8.
     ADDRESS = "2B".
```

```
STATION TD830A:
     DEFAULT = TD8.
     ADDRESS = "1C".
LINE SLCLINE:
     ADDRESS = 7:14:0.
     AUTOPOLL = 15.
     CONTROL = AUTOPOLCTL.
     STATION = TD800A, TD800B, TD830A.
FILE STATIONS:
     FAMILY = DUMMY.
FILE CANDEQUEUE:
     FAMILY = DUMMY.
FILE SUPERQUEUE:
     FAMILY = ALL.
FINI
```

These Transaction Control Language (TCL) statements must be included:

1. PROGRAMBOJEOJ = TRUE

   This statement ensures that code to support the EX (Execute) and HAP (Halt Application Program) commands is included in the generated MCS.

2.
```
PROGRAM A UTILITY:
     TITLE = ODESY/FORMAINT.
     INTERFACE = MCS.
PROGRAM B UTILITY:
     TITLE = ODESY/EDIT.
     INTERFACE = MCS.
     DETACHMESSAGE = TRUE.
```

   FORMAINT and EDIT must be declared utility programs with an MCS interface, and the detach message option must be set to TRUE for EDIT.

3. Station declarations for all stations in the supervisory MCS remote file.

Other TCL statements will be determined by your specific requirements. These are sample TCL statements for a small GEMCOS/ODESY/CANDE network:

```
CONTROL = GENERATE, LIST, COMPILE.
GLOBAL:
     PROGRAMBOJEOJ = TRUE.
     OBJECTCODEFILENAME = SUPER/MCS.
```

```
        SOURCECODEFILENAME = SUPER/MCSSOURCE.
        NCCOKRESPONSE = "****".
        SIGNALCHARACTER = "*".
        MAXTEXTSIZE = 1920.
        MAXRUNNING = 3.
        QUEUENAME = SUPERQUEUE.
BEGIN
        PROGRAM A UTILITY:
             TITLE = ODESY/FORMAINT.
             INTERFACE = MCS.
        PROGRAM B UTILITY:
             TITLE = ODESY/EDIT.
             INTERFACE = MCS.
             DETACHMESSAGE = TRUE.
        PROGRAM C UTILITY:
             TITLE = CANDE.
             INTERFACE = MCS.
        STATION TD800A:
             SCREENSIZE = 1920.
        STATION TD800B:
             SCREENSIZE = 1920.
        STATIONS TD830A:
             SCREENSIZE = 1920.
END.
```

You must execute the supervisory MCS from the ODT. All of the other programs will be zip-executed as and when they are required.

For the sample TCL statements shown above, enter at the ODT:

```
EX SUPER/MCS
```

As soon as the supervisory MCS is initiated the stations will be attached to it and any valid GEMCOS command may be entered.

The commands used to transfer stations to and from ODESY are EX and HAP. Enter:

```
*EX ODESY/EDIT
```

The station will be attached to EDIT. If EDIT is not already operating, it will be zip-executed.

After this point you may enter standard ODESY commands, such as:

/HELLO 1 SUP1 or /STATUS.

The supervisory MCS is non-participating once the station has been attached to ODESY, so response time should not be affected. Only messages preceded by a signal character (*) are processed by the supervisory MCS. To detach the station from EDIT, enter:

/BYE

If EXTRACT is to operate concurrently with EDIT and a supervisory MCS, it must be initiated by the ZIP command, even if executed from the ODT. The mix number is that of ODESY/EDIT. Either enter at the ODT:

<mix number>AX ZIP EX ODESY/EXTRACT

or enter at a control station:

/ZIP EX ODESY/EXTRACT

Supervisory Message Control System (SMCS)

When operating ODESY using SMCS, the remote file STATIONS cannot be declared in the NDL. SMCS assigns its own remote file to the ODESY stations.

NOTE

If the number of remote files to be used at any one time in the network exceeds the number of files declared in the NDL, the MAX FILES statement must be used in the DECLARATION section. For example, if SMCS is to supervise all of the terminals in your network, you only need to declare one remote file (MCSREMOTE) for SMCS's use. You will probably use

more than one remote file in actual production, so you need to set MAX FILES.

The Network Controller may be specified exactly as shown in the GEMCOS example, except the file section should contain only the remote file MCSQUEUE with "FAMILY = ALL".

It is recommended that a jobs file be created before any programs are zip-executed from SMCS. The jobs file is simply a data file containing program names and run time attributes for each program. A jobs file will allow the ON command to be used in attaching stations to ODESY.

To begin system operation enter at the ODT:

EX SMCS

The Network Controller is started up automatically. As soon as SMCS is operating, all stations in the remote file will be attached to it and any valid SMCS command may be entered.

The command used to transfer stations to ODESY is ON. If EDIT is not already operating, it will be zip-executed when you attach the station to EDIT by entering:

ON ODESY/EDIT

After this point, you may enter standard ODESY commands, such as:

/HELLO 1 SUP1 or /STATUS.

To detach the station from EDIT, enter:

/BYE

NOTE

Do not use OFF to detach a station from ODESY. Always use the BYE command.

To attach a station to FORMAINT, enter:

*ON ODESY/FORMAINT

The execution of EXTRACT is the same for SMCS as it is for GEMCOS. Refer to the information discussed earlier in this section.

RECONFIGURATION

Reconfiguration is the process of changing the system configuration parameters. It may consist of adding an operator or station, changing the table sizes of the programs, or changing the date type.

Reconfiguration has three basic steps:

1. Using the Edit Program, enter system maintenance requests into a batch in the Tank file. Batch 999, containing the standard system configuration requests, may be modified or a new batch may be created.

2. Schedule and extract the batch to create a file called ODESYSETUP/INPUT.

3. Process this file using the File Maintenance Program to change the system configuration.

As mentioned, batch 999 contains the standard configuration parameters and may be modified to reflect the desired configuration. Use the formats in Appendix B to enter system parameters.

If you plan to reconfigure during installation, decide what is to be changed before executing the programs.

The reconfiguration process has the following steps:

1. Execute the Edit Program.

2. Log in at an ODESY station: /HELLO 1 SUP1

3. Open batch 999 in Enter mode: /OPEN 999 ENTER

4. Display each system maintenance request and decide which must be changed: /<record>

5. If a system maintenance request is to be inserted, use the INSERT command to insert the new record after the record specified: /INSERT <record> <format>

   Or use the ENTER command to add a new record to the end of the batch: //<format>

6. If a system maintenance request is to be changed, use the MODIFY command: //<record>

7. If an operator or station is to be deleted, use the MODIFY command to change the maintenance action from I to D.

8. Include a Print All maintenance request at the end of the batch so that the complete system configuration will be printed.

9. Close the batch when all changes have been made: /CLOSE

10. Schedule the batch for extraction: /SCHEDULE 999

11. Remove any previous reconfiguration file from disk, such as ODESYSETUP/INPUT.

12. Execute the Extraction Program.

13. End the Edit Program.

14. Execute the File Maintenance Program.

On the first executions of FORMAINT, EDIT, and EXTRACT after reconfiguration, the programs may dynamically change their memory requirements. If this happens, they will go to end-of-job and then either restart automatically, or ask to be restarted.

EXECUTING ODESY

To operate ODESY for the first time, there must be at least one data entry station operating. To use ODESY at a given station, that station must be declared in ODESY's system file by its logical station number (LSN).

The standard system configuration includes eight stations, LSNs 3 through 10. These eight stations are defined as TD 830 type terminals with US and RS as forms delimeters. On the 10.0 MCP release and below, this corresponds to the third through tenth stations listed in your network controller's specification (NDL). This changes on 11.0 NDL. See the NDL 11.0 release documentation.

If your network controller's stations 3 through 10 are not available for use by ODESY, or the terminal types are not compatible, the station parameters in the system configuration must be changed. Do this by executing FILEMAINT with a set of system maintenance requests for station insertions. A "card image" file is required to do this, and it may be created using CANDE, TEXT/EDITOR, or by punching cards and copying them to disk. The card layout is:

COLUMN	DESCRIPTION	VALUES
1-3	Record type	STA
4	Maintenance action	I = insert, D = delete
5-6	Logical station number	Order of the station as defined in the NDL.
7-16	Station name	User defined.
17-20	Station buffer size	TD 700 = 0325 TD 73X = 0480 TD 801/821 = 1920 TD 802/822/83X = 1920
21	Forms set	A = braces B = brackets C = US, RS
22	Control station	C if permanent control station.
23-24	Terminal type	32 = TD 700 33 = TD 730A 34 = TD 730B 41 = TD 801 42 = TD 802 43 = TD 821 44 = TD 822 45 = TD 831 46 = TD 832, MT 983

You may list the file ODESYSETUP/INPUT and inspect the layout of the STA card image. The file created in this way can be processed by FILEMAINT. <input file name> represents the card image file. At the ODT enter:

```
EX ODESY/FILEMAINT; FILE INPUT NAME <input file name>
```

It is important to always check the reports produced by the Filemaint program. The standard system configuration may be printed by entering at the ODT:

```
EX ODESY/FILEMAINT; FILE INPUT NAME ODESYSETUP/PRINT
```

The contents of the Tank file may be printed by executing the PRINT Program. At the ODT enter:

```
EX ODESY/PRINT
```

The report should show one batch, batch 999, which contains data for establishing the standard system configuration.

SECTION 3

FILE MAINTENANCE PROGRAM

The File Maintenance Program (FILEMAINT) is an off-line program which takes a set of system maintenance requests and uses them to update the System file. It may be used to create and initialize the three ODESY master files: the Formats, System, and Tank files. Its main function, however, is to maintain various records, in the System file, that you may alter. Use FILEMAINT to perform these maintenance operations:

1. Modify system configuration parameters.
2. Insert and delete stations.
3. Insert and delete operators.
4. Modify message texts and commands.
5. Insert and delete check digit verification tables.
6. Print all or part of the System file.
7. Squash the Formats and Tank files.

System file maintenance is required during installation and whenever a change is made to the System file.

MASTER FILES

The System file contains details of operators, stations, check digit schemes, system messages, etc. It is the main source of global information about the system. All ODESY programs access this file.

The Formats file contains the screen formats which are used for data entry. Each format describes the screen layout, audit checks, output record layout, and journal layout. These formats control the processes of data entry and extraction. The file contains one special group of formats called SYS. These are the 14 system parameter formats used by ODESY for the entry of system configuration data. They are described in Appendix B.

The Tank file contains the batches of data entered and provides intermediate data storage. The data can thus be easily retrieved for display, verification, and modification. The extraction process takes the data from

the Tank file and converts it into the form required by the application systems. One special batch is used by ODESY. This is batch 999 and it contains the system maintenance requests that correspond with the standard system configuration.

STANDARD SYSTEM CONFIGURATION

The Formats, System, and Tank files comprise a set of standard files called the standard system configuration. The standard system configuration is the collection of system parameters that enable you to operate ODESY for the first time. Once ODESY is operating, you may tailor the system configuration to suit the needs of your particular business by following the reconfiguration procedures.

The standard system configuration contains defined values for the following:

1. Eight stations, numbered 3 through 10, terminal type 46 = MT 983 with a screen size of 80 X 24 characters.

2. Eleven operators, five of which are supervisors, with these operator numbers and passwords:

    1 SUP1
    2 SUP2
    3 SUP3
    4 SUP4
    5 SUP5
    10 OP10
    11 OP11
    12 OP12
    13 OP13
    14 OP14
    15 OP15

3. Standard system messages and commands for data entry and format maintenance.

4. Tank file record size of 180 bytes and maximum input size of 160 bytes.

5. Default table sizes for the Edit Program, Format Maintenance Program, and Extraction Program.

6. USA date type, page size of 60, and percent Tank full of 95.

7. Default restrict and include options.

8. Three check digit verification schemes with these scheme numbers:

   1 -Burroughs Modulus 10/IBM Modulus 10.
   2 -IBM Modulus 11.
   3 -Burroughs Modulus 11.

When FILEMAINT is executed, individual maintenance requests are directed to a workfile called ODESYWORK/<job number> and a line is printed in the File Maintenance Report indicating the action requested. The disk directory is searched to ensure that all necessary disk files are present. If they are present and not in use by another ODESY program, file maintenance is performed. If they are present and in use, only print requests are processed.

FILEMAINT performs only a minimal amount of input parameter checking because it is assumed that all maintenance requests have been entered through and checked by the standard formats supplied with the system.

INITIALIZING FILES

Initialization requires the same input formats used for configuration parameters.

For the Formats and Tank files, the files are created, various pointers are initialized, and directories are cleared. Maintenance of the Formats file is handled by the Format Maintenance Program and maintenance of the Tank file is handled by the Edit Program.

The System file also is created and configuration parameters are inserted. However, further maintenance requests for operators, stations, and message texts are required before the System file is usable. See instructions under "Recovery" in Section 10 for more information on file initialization.

PRINTING FILES

FILEMAINT may be used to print either the entire System file or selected records. Use the PRNT input format, shown in Appendix B to specify the parameters to be printed, or EX ODESY/FILEMAINT FILE INPUT NAME ODESYSETUP/PRINT to print the entire file. To print all or part of the formats file, see the PRINT command in the Format Maintenance Program. To print the Tank file, see the PRINT and DUMP programs.

SQUASHING FILES

The Tank file expands in size as data is entered, but it does not automatically contract in size. Operators may delete batches, but this simply creates free records within the existing Tank file; it does not reduce the size of the file. As new data is entered these free records are used before any records are added to the end of the file. However, if the Tank file has already over-expanded, it is necessary to use FILEMAINT to squash the file. This will move all free records to the end of the file and reallocate the space.

Similarly, deleting a format or group of formats from the Formats file creates free records and the file must be squashed when it over-expands.

To squash either the Formats file or Tank file, or both files, complete a SQUA input format. This input data, when extracted, will create an input file which is then processed by FILEMAINT. Another option is to EX ODESY/FILEMAINT FILE INPUT NAME ODESYSETUP/SQUASH to squash both the Tank and Format files.

Figure 1 shows a sample File Maintenance Report.

```
B1000 ON-LINE DATA ENTRY SYSTEM     F I L E   M A I N T E N A N C E   R E P O R T    TIME 11:41  DATE 01/23/81  PAGE 002
-------------------------------     ---------------------------------------------    ----        ----             ----

CONFIGURATION DATA
MAX STATIONS  CONSTANT TABLE  MAX FORMATS  MAX FIELDS  MAX USER PROGRAMS  MAX INPUT  TANK RECORD SIZE  TANK WARN. MSG. %  PAGE SIZE

    11             0200           06          0100            C0             0160         A  180                95            60

MAX CONST FMNT  MAX SCRATCHPAD FMNT  MAX FIELDS FMNT  MAX FORMATS EXTR  MAX FIELDS EXTR  MAX CONST EXTR  MAX OUTPUT EXTR

     0500            0100                  063               10              0200            1000            400

RESTRICT OPTIONS
   DUP-ANY    ACCEPT     CONTROL    DELETE     INSERT     SCHEDULE  SCH.DEL   EXT:SCH    TERMINATE
    N          N          N          N          N          N         N         N          N

INCLUDE OPTIONS
   CDV        BTCH:TOT  XFOUT      ACCEPT     CONTROL    DELETE    DISPLAY   INSERT     READY      EXT.SCH
    I          I         I          I          I          I         I         I          I          I

   SS         STATUS    WHO        DS
    I          I         I          I
```

Figure 1. File Maintenance Report

FILE SIZES

The physical characteristics of the standard files supplied with ODESY are:

Characteristic	Formats	System	Tank
Record length	60	45	180
Records per block	3	4	1
Records per area	300	480	200
Maximum areas	40	1	100

The size of the System file will accommodate all users, but you may need to increase the sizes of the Tank and Formats files. To do this, modify ODESY/FILEMAINT and either expand the files with DMPALL, or cause them to be recopied by FILEMAINT, such as by doing a squash. You may increase the number of records in the file, but do not change the blocking factor. Always use FILEMAINT rather than DMPALL to change the record size.

For example, to change the limits of the Tank file to 100 areas of 1000 records, modify the File Maintenance Program. At the ODT enter and transmit:

```
MO ODESY/FILEMAINT; FILE TANK BLOCKS.AREA=1000
     AREAS=100
```

When the program has been modified, either squash the file or produce a new file ODESYDATA/TANK with the desired file limits by copying the existing Tank file. Enter and transmit at the ODT:

```
EX DMPALL
```

Then enter and transmit:

```
<mix number>AXPFM DSKDSK ODESYDATA/TANK ODESYDATA/TANK
180 1 1000 100
```

If the file limits are increased by copying the files using DMPALL, the File Maintenance Program must be modified at the same time because squashing the Formats or Tank files, or changing the Tank file record size, creates a new file using the file limits given to FILEMAINT.

When evaluating file limits for the Formats and Tank files, note the following:

1. For the Tank file, each record entered by an operator occupies one record in the file. There is also one extra record for each batch allocated and another 103 records for system use. The Tank file needs to be large enough to contain the following number of records:

   103 + <number of batches> + <number of data records>

2. For the Formats file, there is one record for each field, an extra record for each field with constants, and one record for each format header. There are also 799 records for system use. The Formats file needs to be large enough to contain the following number of records:

   799 + <number of formats> + <number of fields> + <number of fields with constants>

The blocking factor for the Tank file must not be changed because the on-line version of the Extraction Program and the recovery system of the Edit Program depend upon unblocked records.

SECTION 4

SYSTEM PARAMETERS

The ability to change the system parameters in ODESY provides a simple means of altering the system as your requirements change. Most of the flexibility in ODESY is due to the fact that information is held on disk as system parameters, not compiled into the programs.

Most of these system parameters are stored in the System file. They include information about operators, stations, system messages, check digit verification schemes, page sizes, dates for reports, and table sizes for programs. These system parameters are changed by the File Maintenance Program discussed in Section 3.

The File Maintenance Program is an off-line program which processes a serial file containing System file maintenance requests. This file can be obtained by using the data entry facilities of ODESY to enter the system parameters through an on-line terminal display unit and then producing a serial file on disk by executing the Extraction Program.

The formats which are used to enter the system parameters are listed in Appendix B. The group name is SYS and the individual format names are shown. Data corresponding to the standard system configuration is in batch 999. This batch contains all of the standard system messages, commands, and check digit verification schemes. To reconfigure the system, enter new data into or modify batch 999, extract the batch, and use the resulting file as input to the File Maintenance Program.

SYSTEM CONFIGURATION PARAMETERS

System configuration parameters consist of parameters for the Edit, Extraction, and Format Maintenance Programs. Most affect only one program, but some system configuration parameters affect several programs.

Date type is used by all programs which print reports. Date type determines whether the date is printed in USA format (MMDDYY), European format (DDMMYY), or International format

(YYMMDD). It is also used by the Edit Program to give the system default date type for checks on date fields.

Page size is also used by all programs which print reports. It allows ODESY reports to conform to a particular size of paper.

EDIT CONFIGURATION PARAMETERS

These parameters are used by the Edit Program at the time of execution to determine the sizes of certain tables held in dynamic memory. Dynamic memory means that table sizes are under the control of the program at run time, so it is not necessary to recompile the programs to alter table sizes. Simply change these parameters.

Standard System Configuration Values

The first three parameters, Maximum Formats, Maximum Fields, and Maximum Characters of Constant Data determine the sizes of the format table, field table, and constant table. These values are dependent on each other in this way: if the field table has room for the fields of five formats and the constant table has room for the constants of three formats, it is useless to have more than three entries in the format table, because only three formats can be in memory at a time. As long as the Edit Program has room for at least one format with all of its fields and constants, data entry can continue. However, if this is the case, each new format required must be loaded from disk. With larger table sizes more formats can be held in memory and there is less need to access the disk.

Maximum User Programs controls the number of user programs which may be in execution at a time. If the user program table fills to capacity, ODESY informs the operator so that this parameter can be changed. If there is never more than one user program executing, give the option a value of one. If inquiry facilities are not required, give the option a value of zero.

The Shutdown option determines when an inquiry program is sent an End-of-file message. It also determines how the inquiry program is informed when its last batch is closed.

Maximum Input Size determines the size of the data area allocated to each station for the current data record. Derive this value from the largest input size of any format in the system, as shown in the Format Maintenance Report.

Percentage for Tank File Warning determines the point at which the Edit Program gives a warning that the Tank file is nearly full. The warning is given when the percentage of data records in use is greater than the parameter.

Tank File Record Size controls the length of a record in the Tank file and also controls the largest number of characters accepted from one format during data entry. Three values are possible for Tank File Record Size, but the parameter must be at least 20 bytes larger than the maximum input size. The smallest value, 180 bytes, is suitable for a maximum input size of 160 bytes or less. The size 540 bytes suits a maximum input size of 520 bytes. The largest value is 1080 bytes, allowing space for 1060 bytes of input data. Note that if you change this value the entire Tank file is copied to form a smaller or larger Tank file. If you increase the record size no data is lost, but if you decrease it any data in excess of the new Tank file size is deleted.

The option also has an effect on the maximum number of stations. A recovery record is written to the Tank file containing information about each station, so more stations may be handled with a larger record size. For instance, if 25 stations were required, the option would have to be set to B, even if 160 characters of input data per format were sufficient.

This table shows the relationships between various values:

Option Setting	Actual Tank Record Size	Maximum Value of Input Data Size	Maximum Number of Stations
A	180 bytes	160 characters	17
B	540 bytes	520 characters	56
C	1080 bytes	1060 characters	99

Response Time Considerations

Response time depends on your particular working environment. In general, high parameter values produce better response time, but use more memory. Low parameter valuesuses less memory, but produces poorer response time.

Initially, use the parameters in the standard system configuration distributed with ODESY. If response time is poor, it may be useful to increase the values of the parameters. If response time is good, but you consider memory usage to be too high, decrease the values of the parameters.

To estimate optimum parameter values, consider which formats are likely to be in use at any one time. For example, there may be 10 stations in the network, but most of the time they are used to enter invoicing data requiring three main formats. The Format Maintenance Report shows that these three formats contain 95 fields and 267 characters of constant data. Suppose that there are another six formats which are used less frequently and it would be convenient to have space for three out of the six. This means another 50 fields and another 200 characters of constant data.

On this basis, Maximum Formats should be six, Maximum Fields should be 150, and Maximum Characters of Constant Data should be 500. In practice, it is preferable to increase Maximum Formats to eight, allowing a larger number of small formats. As shown, the number of stations is unimportant.

Test this set of values by executing EDIT and measuring response time. Then vary the parameters and assess the differences caused by various values. The critical parameter is Maximum Fields; it has the greatest effect on memory requirements. Determine the other parameters after finding the best value for this option.

FORMAINT CONFIGURATION PARAMETERS

These parameters are used by the Format Maintenance Program at run time to determine the sizes of certain tables held in memory. They are equivalent to the parameters which affect the Edit Program, but have one additional effect which the other parameters do not have. Because all formats are

entered using the Format Maintenance Program and because they control the number of fields and constants handled by FORMAINT, they also control the maximum number of fields and constants that can be in any format in the system.

Maximum Fields determines the size of the field table. The Format Maintenance Program only has one format in memory at a time, so this parameter also limits maximum fields per format.

There are two constant tables and two parameters to determine their sizes. The main constant table holds all field names, inserted constants, range check constants, and value check constants for all fields in the format. It provides a limit on the maximum number of characters of constant data per format. The scratchpad constant table is used to hold the constants for each field as it is amended, inserted, or listed. Therefore it provides a limit on the maximum number of characters of constant data per field.

Some constants are needed by the Edit Program but not by the Extraction Program, and vice versa. Some fields, such as output only fields, are needed by the Extraction Program but not by the Edit Program. The table requirements of each format by program are listed in a Format Maintenance Report.

EXTRACT CONFIGURATION PARAMETERS

These parameters are used by the Extraction Program at run time to determine the sizes of certain tables held in memory. They are equivalent to the parameters already discussed in this section for the Edit Program.

The first three parameters, Maximum Formats, Maximum Fields, and Maximum Characters of Constant Data are dependent on each other. They determine the sizes of the format table, field table, and constant table, and they should be derived in the way parameters are determined for EDIT.

The formats held in the tables at any one time belong to one group, so the sizes of the format and field tables probably are smaller than those for EDIT. Determine which formats in each group will be in regular use, and give values to the first two options according to the size of the largest group. Determine the number and size of output only fields

used by formats and set the maximum characters of constant data option accordingly. Note that if low values are given to the parameters, more disk must be accessed to load the formats which are not in memory.

Maximum Output Record Size determines the size of the output buffer used to write records to the application file. It can be derived from the record sizes of the files created by ODESY.

DUPLICATE ANY PARAMETER

Duplicate Any is used by the Edit Program to aid in controlling the duplication of data from format to format during data entry. If this option is set, data may be duplicated from the previous format regardless of whether the format name has changed. The default option allows data to be duplicated from the previous format only if the format name is the same.

RESTRICT CONFIGURATION PARAMETERS

The Restrict Options enable the use of certain EDIT commands to be restricted to supervisors, providing greater control over the data entry function. These parameters are used with the supervisory status option discussed later in this section.

A command can either be restricted for use only by supervisors or unrestricted for use by any operator. The commands that may be restricted are ACCEPT, CONTROL, DELETE, EXTENDED SCHEDULE, INSERT, SCHEDULE, and TERMINATE. In addition, you can restrict the delete options of the SCHEDULE command. If the SCHEDULE command is unrestricted, but the schedule delete option is restricted, all operators can schedule batches for extraction but only supervisors can schedule them for deletion.

INCLUDE CONFIGURATION PARAMETERS

The Include Options enable certain ODESY functions and commands to be disabled during data entry, providing additional control over data entry.

The functions which may be excluded are check digit verification, batch totalling, crossfooting, and on-line format maintenance. If a feature is excluded, ODESY ignores any reference to it in a format or command.

The commands which may be excluded are ACCEPT, CONTROL, DELETE, DISPLAY, DS, EXTENDED SCHEDULE, INSERT, SS, STATUS, and WRU. If ACCEPT is excluded, an operator is not allowed to force entry of a record containing errors. If CONTROL is excluded, only permanent control stations and the ODT are allowed to use the TERMINATE and ZIP commands. If SS is excluded, operators are prevented from sending messages to each other.

STATION PARAMETERS

You must define the characteristics of all ODESY stations in the network by entering parameters for each station.

Logical Station Number is the terminal number assigned by the NDL to identify the station. It is derived from the station section of the NDL listing by the order in which the stations are declared. The first station is number 1, the second station is number 2, etc.

Note that this changes on NDL release 11.0. See 11.0 NDL release documentation.

Station Name is for documentation purposes only. It is obtained from the NDL listing.

Station Buffer Size is used by ODESY for data communications purposes. See the table in Section 2 under "Executing ODESY" for buffer sizes. Forms Set defines whether brackets or braces are to be used as forms delimiters on the TD 700 or TD 800. For the TD 730, TD 820, or TD 830, ODESY uses US and RS as forms delimiters.

Terminal Type is used by ODESY for data communications purposes to control several functions. Obtain this value from the station section of the NDL listing. It is very important that this option be set correctly.

Control Status, when set, gives a station the power to use the CONTROL, TERMINATE, and ZIP commands during data entry.

OPERATOR PARAMETERS

You must define the characteristics of all ODESY operators by entering parameters for each operator.

Operator Number, entered during log-in, identifies the operator and appears in all reports to indicate who is working at a particular station on a particular batch.

Operator Name is for documentation purposes only.

Operator Password, entered during log-in, provides a security check to prevent access by unauthorized operators.

Operator Format Range is another type of security check. ODESY enables the operator to use a format only if the security level of the format is within the range specified for the operator.

Operator Type, when set to supervisor status, gives an operator the power to use certain restricted EDIT commands. Use this option in conjunction with the restrict options.

MESSAGE TEXT PARAMETERS

Default message texts used in the system messages for the Edit Program and the Format Maintenance Program are included in the standard system configuration distributed with ODESY. Enter system message parameters only to make changes to the default message texts.

Text Number identifies the system message. Messages numbered 1 through 99 are used by the Edit Program and those numbered 101 through 199 are used by the Format Maintenance Program.

A system message is displayed on the screen in response to input from the operator. Most system messages are 31 characters in length, but several have fewer characters, as stated in the explanation of each message.

The system messages shown in Appendix D pertain to the Format Maintenance Program. Messages pertaining to the Edit Program are documented in the B 1000 Series On-Line Data Entry System (ODESY) Terminal Operator's Manual, form 1131851.

COMMAND PARAMETERS

Only the first three characters of each command used during data entry and format maintenance are recognized by ODESY. Similarly, keywords used to further specify a command are also identified by their first three characters. For instance, while executing the Format Maintenance Program, rather than entering /AMEND FORMAT, it is necessary to enter only:

```
/AME FOR
```

ODESY interprets this input as a request to change a format header.

Default 3-character abbreviations are included in the standard system configuration distributed with ODESY. Enter command parameters only to make changes to the default abbreviations. Note that every field must contain an abbreviation; if you change any abbreviation, all data on that format, either default values or new values, must be transmitted.

No two EDIT or FORMAINT commands may have the same abbreviation; ODESY is unable to distinguish between them.

CHECK DIGIT SCHEME PARAMETERS

Check digit verification, one of the audit check features provided with ODESY, is the process of re-evaluating a check digit and comparing it with the check digit entered. A check digit verification scheme converts a number into a

self-checking number by adding an extra digit, called the check digit, that is calculated from the other digits in the number. Check digit calculation calculates the check digit, based on a modulus, a remainder, and a set of weights.

Check digit verification enables you to detect nine of 10 mistakes made with numeric data and is especially useful in detecting the interchange of two consecutive digits. To use check digit verification, you must have some method of generating the check digits in the first place. For this reason, check digit verification is most successful with numbers which are used regularly and are relatively stable, such as employee numbers, customer numbers, and account numbers.

ODESY enables you to define a maximum of nine check digit verification schemes. The standard system configuration distributed with ODESY uses the following schemes:

1. Scheme 1 is defined as Burroughs Modulus 10 (same as IBM Modulus 10).

2. Scheme 2 is defined as IBM Modulus 11.

3. Scheme 3 is defined as Burroughs Modulus 11.

The parameters used to define the three default schemes are listed in Appendix B.

For a detailed explanation of check digit verification schemes refer to Appendix C.

Scheme Number identifies the check digit scheme. The scheme number must be specified when a field requires a check digit verification check, as shown in Section 5.

Modulus and Remainder are used in the calculation of the check digit. Modulus 10, modulus 11, and modulus 13 are frequently employed.

Calculation Type specifies whether weights or products are to be entered, and if weights are entered it also indicates whether all digits are to be added.

Complemented specifies whether the final product is to be subtracted from the modulus in calculating the check digit.

Weights or products provide a balanced distribution of check digits so that similar numbers will not have the same check digit. ODESY uses a table access method to handle check digits. It holds products in the tables, so when weights are entered, ODESY calculates the corresponding products. Appendix C shows examples of both weights and products.

PRINT PARAMETERS

Print parameters are used to request printing of all or part of the System file. If an operator is being inserted, or a station is being deleted, or a system message is being changed, the File Maintenance Report shows the effect of the maintenance request, but in order to produce a list of all operators, stations, or system messages it is necessary to enter print parameters.

The most common print parameter is ALL, which produces a listing of the entire system file. The CDV, CMD, OPR, STA, and TXT parameters produce lists of check digit schemes, commands, operators, stations, and system messages. The CON parameter produces a list of all system configuration parameters. The CTL parameter summarizes the data in the control records of the three master files.

A maintenance request to print the entire System file is included with the release. This file is called ODESYSETUP/PRINT and may be used as input to FILEMAINT.

SQUASH PARAMETERS

Squash parameters are used to request squashing of the Formats file and/or the Tank file.

As data records are no longer needed, they are changed into free data records scattered throughout the file, but the disk space allocated to the file is not reduced. For example, the Tank file may be 10,000 records long, but of these only 4000 are active data records and the rest are free records. The file is this large because at one time it did contain 10,000 records and though batches have been

deleted since that time, the disk space remains allocated. Squashing the Tank file produces a file containing just 4000 records.

In general, the Tank file will require squashing more frequently than the Formats file, because the latter is usually much smaller and the size of the Formats file is likely to be more stable than the size of the Tank file.

A maintenance request to squash both the Formats file and the Tank file is included with the release. This file is called ODESYSETUP/SQUASH and may be used as input to FILEMAINT.

SECTION 5

SCREEN FORMATS

Screen formats are the key to efficient data entry. Data entry is the transcription of data from original documents, normally pre-printed stationery, to a medium suitable for input to a computer. The pre-printed stationery, defined over a period of time, is not easy to change. Similarly, the format of the data supplied to the computer, determined by the application program which processes the information, is very difficult to change. The screen format must satisfy the requirements of both the pre-printed stationery and the application program, and be adapted to the needs of data entry operators.

This section describes the format options available and how they may best be used. Format maintenance, the process of actually entering or changing the format data in the Formats file, is discussed in Section 6.

Each format is made up of a number of fields that are referenced by a field sequence number during format maintenance. These numbers determine the field's position on the screen. Each field contains four main types of information:

1. Display options that control the screen layout.
2. Audit check options that determine which checks are to be applied to each field.
3. Output options that determine the characteristics of the application file to be produced.
4. Journal options that describe the printed journal.

The display options and the audit check options are used during data entry. The output options and the journal options are used during extraction. The format contains all of the information that the Edit and Extraction Programs require to process a particular record.

Formats are gathered together in groups. To reference a format, specify both its group name and its format name. A set of group header options is associated with the group name. These options describe the application file to be produced. In other words, each group of formats is used to create just one application file.

A set of format header options is associated with a format name. They are used to describe the features that apply to the format as a whole.

For a complete list of format options, see the second part of Appendix B.

DISPLAY OPTIONS

The display options control the screen layout. The screen layout is the visual representation of an ODESY screen format. It should be designed with consideration for the operators who will use it. Design the screen layout to be similar to the printed form with which operators are already familiar.

FIELD SEQUENCE NUMBERS

Sequence numbers determine the order of fields on the screen and are usually numbered 10, 20, 30, 40, etc., to allow for insertions. Note that column headings and titles do not have entry fields, but still require sequence numbers.

FIELD NAME

Field name is used to describe or explain an entry field. The description or prompt appears to the left of the field during data entry. It may be a maximum of 28 characters long and is truncated at the right; trailing blanks are omitted to give its actual length. Some fields may have no name and other fields may have a name but no data. The latter are described as name-only fields, and are used as column headings and titles on the screen. If a title is longer than 28 characters, two or more name-only fields are required.

FIELD LENGTH

Field Length is the length of the field on the screen. Length restrictions are 15 for numeric fields, 16 for signed numeric fields, and 28 for fields with constants. The maximum for all other fields is 160 characters. A name-only field has a length of zero.

If Field Type is set to total, the batch total specified will be displayed in the field. If Field Type is set to output-only, this field does not appear on the screen. This feature is described later in this section.

The length of an unpacked, signed numeric field with an overpunch sign convention is reduced by one character in the output file. This is because a separate character must be allowed during data entry for the sign. However, that extra byte is not needed at Extract time since the sign is overpunched. The length of a packed field is reduced by exactly half in the output file, whether signed or not.

SPACES BEFORE/SPACES AFTER/LINES AFTER

Spaces Before, Spaces After, and Lines After are used for horizontal and vertical spacing between fields. The spaces before and spaces after options allow for spacing between the previous field and the field name, and between the field name and the field. If it is not a name-only field, there must be at least one space between the name and the field itself. This space is reserved for highlighting on the TD 730 and TD 830. Note that if secure video highlighting is specified, there must be a minimum of two spaces between the field name and the field. This is to allow a space for the blink character in the event of an error in entry and prevents the secured data from being exposed when the error is returned. Lines After is a field option which allows blank lines between fields.

HIGHLIGHT

Highlight allows the use of bright video, underline video, reverse video, and secure video on a TD 830 and secure video on a TD 730. If a screen format with highlighting is used on a TD 700, TD 800, or TD 820, ODESY automatically suppresses the highlights.

DUPLICATE-IN

Duplicate-in allows data to be duplicated from screen to screen as long as the format does not change. (A system configuration option, Duplicate Any, may be set to allow data to be duplicated from screen to screen even if the format does change. Refer to Section 4.)

The first time a format is required, the duplicate-in fields must be given initial values, but from then on the values are duplicated from record to record. The fields are duplicated as default values so that the operator can easily change them. This feature is different from the duplicate-out and output-only options because the duplication is visible to the operator at all times.

Inserted Constant allows an automatic default value to be given to a field. It can be used with the duplicate-in option to give the default value for the first record using the format.

JUSTIFY IN

Justify In has one important effect on screen layout: when the option is set to R and the terminal is a TD 730, TD 820, or TD 830, the right-justify feature of the terminal is used to automatically justify the data on the screen.

VERIFY METHOD

This option allows data within a field to be verified in one of four ways. The first two options require that the data be re-entered manually. The difference between these two options involves what is displayed when the field is in error. With the default rekey option, the data keyed most recently is displayed in the error field. With the second option, the data keyed initially, such as during data entry, is displayed.

The third option, display verify, causes the data to be displayed on the screen during verification. For display verify, the cursor will still tab through the data field. The final option, no verify, causes data to be displayed on the screen during verification and also makes the field

transmittable-protected (for terminals that support this function). This options is useful if there are display verify fields that you do not wish to tab through manually. If you choose this method, the cursor tabs over the field and the field data cannot be changed during verification. If this feature cannot be used on your terminal, ODESY will treat these fields as display verify fields.

For the first three options, the error fields are blinked (on terminals that support the blink feature). Error fields are simply filled with question marks on terminals that do not support the blink feature.

FORMAT TYPE

Format Type, a format header option, may also have an effect on screen layout. If this value is set to S or U, the user program is requested to supply data for the record being entered, and the data supplied is placed within the format as default data for the operator. Inquiry formats are discussed further in Section 9.

A value of D in Format Type allows the display of batch totals on a user-formatted screen. This non-data entry format can be displayed at any time, including in display or verify modes, for checking the current status of batch totals.

AUDIT CHECK OPTIONS

Audit checks are the checks performed every time a data record is entered, verified, modified, or inserted (or inquired upon in Inquiry mode). ODESY offers most of the standard audit checks as options.

FIELD TYPE

Field Type allows the specification of five field types:

1. Numeric field. This may contain only the digits zero through nine.

2. Signed numeric field. This may contain an optional leading sign, either plus or minus, followed by the digits zero through nine.

3. Alphanumeric field. This may contain anything.

4. Output-only field. This does not appear on the screen.

4. Total field. This displays the specified batch total.

INSERTED, RANGE, AND VALUE CONSTANTS

Inserted constant, range check constant, and value check constant all require at least one constant. Only one of the three options may be selected. In each case the first constant is used as a default value and is displayed on the screen.

Range Check applies only to numeric and signed numeric fields. Three constants are required. The first constant is used as a default value and is displayed on the screen, and the second and third constants are the lower and upper limits of the range. The first constant may be blank. The field is in error if the value entered for the field is outside of the range specified.

Value Check may be applied to numeric, signed numeric, or alphanumeric fields. It requires at least one constant. The first constant is used as a default value and is displayed on the screen. It may be blank. If a blank default value is required but blank is not an acceptable value, set the mandatory check option. All constants, including the first, are compared with the value entered for the field and the field is in error if a match is not found. The only limit on the number of constants in a value check list is the size of the scratchpad constant table.

MANDATORY CHECK

Mandatory Check is used to force the entry of data into a field. The field is in error if no data is entered.

DATE CHECK

Date Check allows the checking of 5-digit Julian date fields, or 6-digit date fields of DDMMYY, MMDDYY, or YYMMDD format. The format is determined by the date type option, which is a group header option, so it remains fixed for all of the fields of a particular application file. A date such as February 29, 1981 is detected as an error.

CHECK DIGIT VERIFICATION

Check Digit Verification (CDV) applies only to unsigned numeric fields and allows check digit verification by a number of schemes. You may define a maximum of nine check digit verification schemes. Three common schemes are included in the standard system configuration distributed with ODESY. Check digit verification schemes are discussed in Appendix C.

CROSSFOOTING

The Crossfooting option (XFOOT) applies only to numeric or signed numeric fields. Crossfooting is a form of subtotalling. Fields within a format can be added to or subtracted from the crossfoot register in such a way that the subtotal balances against the sum of the values for that format. Normally the fields being totalled are added and the total itself is subtracted so that a correct crossfoot register will contain a value of zero. The crossfoot options allow each field to be added to or subtracted from the crossfoot register. When a format which uses crossfooting is transmitted during data entry and the crossfoot register is not zero, an error is returned.

BATCH TOTALLING

The data entry program provides ten registers per batch that can be used to accumulate totals over all the records in the batch. A numeric field can be added or subtracted from a specified register. The registers can also be incremented unconditionally or if the field processed is not blank. If this is a batch header format, the record contains operator-entered batch totals and does not contribute to computed totals. When the batch is closed or scheduled, EDIT compares the accumulated totals with the batch header

totals. If an error is detected, a batch total comparison error is reported. An error is also reported if the accumulated totals are not zero and a batch header record has not been entered.

JUSTIFY-IN

Justify-in may be used to ensure that the data entered is right-justified or left-justified correctly. The right-justify feature of the TD 730, TD 820, and TD 830 is used to automatically right-justify where specified; otherwise the data must be right-justified manually. A justify-out option allows unjustified data to be correctly justified for the output file.

OUTPUT OPTIONS

Output options are used to specify the characteristics of the application file being produced.

PHYSICAL CHARACTERISTICS

The physical characteristics of the application file are described by group header options since all formats within a single group produce data for the same file. The physical characteristics of the file must satisfy the requirements of the application system and values can be given to most of these options by referring to the specifications of the application system.

File Name

File Name is the name of the application file produced by ODESY for all batches entered using this group. If a disk file of the same name already exists, ODESY will add records to the end of that file rather than create a new file. The format of the file name varies from disk to tape. For disk files, the family name may be a usercode. See Section 10 for more information on usercode security, and Appendix B for details on specifying files to ODESY.

Medium

Medium indicates whether the type of file to be produced is disk or tape.

Record Size/Blocking Factor

Record Size and Blocked (blocking factor) apply to all media. These values should be defined by the application system. Ensure that the maximum output size for the Extraction program is large enough for all files being produced. Refer to Section 4.

Records per Area/Number of Areas

Records per Area and Number of Areas apply only to disk files. The values may be defined by the application system. Alternatively, the default values are 500 and 40.

Label Type

Label Type applies only to tape files. Refer to Appendix C for details. The default label type, Burroughs standard label, is suitable for most purposes.

RECORD LAYOUT

A record layout which satisfies the application system is usually the final product of ODESY. Find values for most of these options by referring to the specifications for the application system.

Output Position

Output Position defines where the data for the field is to be placed in this record of the application file. The units are bytes and the start of the record is byte 1.

Duplicate-out

Duplicate-out allows data to be duplicated to output records in the application file. For complete details on the use of this option, refer to Write Options and Write Option examples later in this section.

Justify-out

Justify-out applies to all field types (numeric, signed numeric, alphanumeric, and output-only fields) and it allows data to be justified correctly for output. Right-justification should be performed on all numeric data, either by setting the justify-in option or by setting the justify-out option.

Packed

Packed applies only to numeric and signed numeric fields. It enables numeric data to be converted into 4-bit packed (hexadecimal) storage for output. Because the output position is specified in units of eight bits, the packed-out option allows a packed numeric field to start either at a byte boundary as is usual or four bits later at a half-byte boundary.

The length of the field is derived from the field length option, but also depends upon the field type option, the packed-out option, and the sign convention option. A signed, unpacked, numeric field of length L with an overpunch produces a field of length L minus one bytes in the application file. In all other cases, a field of length L produces a field with an output length of L, that is L digits for packed numeric fields which equals L divided by 2 bytes, or L bytes. L is the value given to the field length option.

Sign Convention

Sign Convention is a group header option. All signed numeric fields within one application file must use the same sign convention. There are four sign conventions. For unpacked numeric fields the sign can be leading or trailing, and it can be overpunched or a separate character. For

packed numeric fields the sign can be leading or trailing, so the third and fourth alternatives default to the first and second. The plus and minus signs are represented by hexadecimal C and D in overpunched signs and packed fields. When the sign is a separate character, it is a displayable "+" or "-" character.

During data entry, a sign entered by an operator must be in the leading, or most significant, position regardless of the sign convention. If no sign is entered, one character in the field still must be left blank because at extract time the default sign will be added. (If the sign will be overpunched at extract time, the field length will be reduced by 1. Therefore, even for overpunched sign conventions, signed numeric fields cannot be completely filled with numbers by the operator.)

Blank-fill

Blank-fill applies only to numeric and signed numeric unpacked fields. If blank-fill is specified, areas of the field which are not entered by the data entry operator will be extracted as blank characters. If blank-fill is not specified, the field will be zero-filled at extract time, unless no data is entered into the field. In this case, it is blank-filled by default. Packed fields are always zero-filled, and all other field types are blank-filled by default.

Sign

The Sign option applies only to signed numeric fields. It represents the default sign of a field. A sign need not be entered in the sign field. If nothing is entered in this field, the default is positive. The sign field will be used only if no sign is entered in the data field during data entry, so an explicit sign entered in a signed numeric data field will always override the default sign field.

Output-only

Output-only fields are a special feature enabling data to be placed in the application file without being viewed by the operator. An output-only field must have an inserted constant, representing the data that is to be placed in the

record. None of the audit check options apply to an output-only field and the screen formatting options are ignored because the field is not displayed. However, all output and journal options may be used, and a field name can be entered for printing in the journal.

WRITE OPTIONS

Write options control the writing of records to the application file, enabling you to take advantage of the multiple record features. These features enable you to produce multiple records in the application file from one format or, alternatively, to condense the data from several formats into a single record in the application file.

The multiple record features are implemented in the write-after option and the duplicate-out option, but in order to understand how they work it is necessary to explain how the data in a batch is processed during extraction.

The extraction process uses two record buffers to build up data for the application file. The first one is the output buffer and the second is the duplication buffer. The duplication buffer holds any data that is to be duplicated from record to record and the output buffer contains the data for the current record of the application file. At the start of the extraction of a batch, both buffers are cleared; in other words, blanks are placed in the buffers. The next step is to read the first record of the batch and to load the corresponding format. The data in that record is then processed, one field at a time, in field sequence number order.

For each field in the format, the data is transferred from the batch record to the output buffer and the data is converted if necessary. If the duplicate-out option is set, the data is placed in the output buffer and also in the duplication buffer. If the chained option is set, the data is placed in the output buffer and if the data is not all blank it is also placed in the duplication buffer.

Write-after

Write-after is the last option to be used during extraction before the next field is examined. If the write-after option is not set, no action is taken and the data for the current record remains in the output buffer. However, if the option is set to write-only or write-and-clear, the contents of the output buffer are written as a new record in the application file. For write-only, the output buffer is written and the duplication buffer is copied into the output buffer, causing data to be duplicated into the next record. For write-and-clear, the output buffer is written and both buffers are then cleared, cancelling the effects of duplication.

WRITE OPTION EXAMPLES

It may be useful to give some examples at this point. The simplest possible example is where each format corresponds with one record of the application file. In this case you only need to set the write-after option of the last field of the format to write-and-clear.

A more complicated example is where each record of the application file is built from three formats. In this case the write-after option of the last field of the third format in the sequence should be set to write-and-clear.

Another example is where one format produces four records in the output file, and each record consists of duplicated data such as record type and date, followed by specific data such as quantity and cost. The duplicated data should appear once in the format and the duplicate-out option should be set for those fields. The specific data should appear four times in the format and, for the last field of each set of specific data, the write-after option should be set to write-only. Write-and-clear can be used for the final field of the format if the buffers are to be cleared.

The final example is where one format produces four records in the output file and each record consists of an account number, which may either be duplicated from record to record or changed at any time, followed by specific data such as quantity and cost. This use of the duplicate-out feature may be referred to as a "chained" field. The chained field should appear four times in the format, each time being

specified as chained and followed by the specific data fields for the record. Whenever an account number is entered in the chained field, it will be placed in the duplication buffer and duplicated in the output record until the next account number (next non-blank entry) is entered in a succeeding chained field. For the last field of each set of data, the write-after option should be set to write-only. Write-and-clear can be used for the final field of the format if the buffers are to be cleared.

The example above provides a very efficient way of entering data with ODESY. The amount of data entered by the operator is reduced because the duplicated data is entered once. The overall transmission time is reduced because there is more data in each ODESY record and therefore fewer records.

Write-after can also be set to "save." Refer to the third example. This format produces four records in the application file each time it is used. However, if the operator only has data for two records, ODESY does not produce two good records followed by two records containing only the duplicated data and blanks for the specific data. The output buffer is not written if only blanks have been processed since the last write. The data in the duplication buffer does not count, so only two records are written.

However, if each of the sets of specific data contains an inserted constant, such as "1," "2," "3," or "4" to identify each of the four records, ODESY is not processing blanks, and four records are written. The problem may be overcome by setting the write-after option to save for each of the inserted constant fields. This causes these fields to be treated as blank when determining whether to write a record. The third and fourth records are written only if data is entered into a field other than the inserted constant fields.

JOURNAL OPTIONS

Journal options are used to describe the printed journal. The journal is an optional feature. It is produced by the Extraction program at the same time the application file is produced and is an audit trail of the data extracted. The Extraction program also prints an Extraction Report, summarizing the batches processed and containing a single line for each batch.

The layout of the journal is under your control. It can contain one or more lines for each batch record and the data printed can be all fields or only selected fields.

JOURNAL

Journal is a group header option and it must be set if a journal is required.

NAME COLUMN

Name Column indicates where the field name is to be printed on the print line, from column 1 through column 132. If nothing is entered for the name column option, the field name is not printed in the journal. This option is not dependent on the data column option.

DATA COLUMN

Data Column indicates where the data is to be printed on the print line. The first column is column 1 and the end of the line is column 132. If nothing is entered for the data column option, the field is not printed in the journal.

PRINT-AFTER

Print-after corresponds with the write-after option. For most purposes it is sufficient to set the option to print-and-clear whenever a complete print line has been assembled, but the same facilities available for writing are available for printing, so you may use duplicate-out, print-only, and save.

JOURNAL INPUT AND OUTPUT RECORD NUMBER

Journal Input Record Number and Journal Output Record Number enable you to relate the printed journal to the records in the ODESY batch and application file. They are group header options and if one or both of the options is required, the number entered indicates a column number in the journal. The input record number is the number of the record in the ODESY batch and, if this option is selected, the

corresponding input record number is printed on every line of the journal. Similarly, the output record number is the number of the record in the application file. If you use these options, do not print other data in the columns for record numbers.

JOURNAL

If Journal has been set, you can decide at the time of execution whether to produce an application file, a journal, or both. The Extraction program provides the options of a journal-only execution or a no-journal execution in addition to the standard execution which produces both journals and application files. A journal-only execution would enable you to verify that the correct data was being extracted before producing the application files.

FORMAT HEADER OPTIONS

The remaining options are format header options.

NEXT FORMAT

Next Format provides an automatic sequence of formats. Often the next format is the same as the format name.

SECURITY LEVEL

Security Level is used to give a security level from 0 through 99 to the format. It is checked against the operator's format range whenever the format is requested during data entry. The operator is refused access to the format if the format's security level does not fall within the operator's format range.

NO VERIFY

No Verify is used when a format contains data which does not need to be verified. If a no-verify format is requested, the record is marked as verified as soon as it is entered, so it does not count as an unverified record when the batch is closed.

LINE ADVANCE

Line Advance allows blank lines between the control line information and the first field.

CURSOR ADVANCE

Cursor Advance allows the cursor to be positioned beyond the first field during data entry. For instance, if the inserted constant and duplicate-in options are used to give default data to the first five fields, this option can be set to 5 so that the cursor is automatically placed in the sixth field. Name-only fields and output-only fields are not counted when calculating how far to advance the cursor.

FORMAT TYPE

This option indicates the purpose of the format.

A normal data entry format is created by setting this option to space.

A format of type B indicates to EDIT that the format is used to enter batch totals. When a batch is closed, the data entry program compares the totals accumulated with the totals stored through this type of format. Only one format type B record can be entered in a batch.

A format of type D indicates that the format will not be used to enter data, but to display information to the operator. The display formats are the only type of format that can be requested in verify or display mode. This type of format can be used in conjunction with the field type option T to display the accumulated batch totals.

If audit checks are required which cannot be specified by other options, they may be performed by a user program. If Inquiry Format Type is set to C or U, the data entered is sent to the user program for further checking. If data will be supplied from a data file to the operator, such as for inquiry purposes, an S or U format type should be specified. To send data which does not require checking to a user

program, a K format type should be specified. Inquiry formats are discussed in Section 9.

SPECIAL FORMATS

ODESY uses two sets of special formats and they are shown in Appendix B.

The formats for the File Maintenance program have group name SYS and have format names CONF through SQUA. They are used in batch 999, which contains the current ODESY configuration. System parameters are described in Section 4.

The formats for the Format Maintenance program are contained within the program and are not changeable. They are used during format maintenance to describe the format data being entered. Included are two group header screens, one format header screen, three field details screens, and one screen for each of the field check options: constant, range, and value. Format maintenance is discussed in Section 6.

DESIGNING A FORMAT

To translate a file and record description into an ODESY format, use the Field Screen Layout/Audit and Output Specifications Form contained at the end of Appendix B.

1. Obtain the file specification and record layout to establish required output.

2. Decide what is to appear on the screen. For instance, should the format be designed to produce several output records? Are there to be fields which appear in the output record but not on the screen?

3. Design the basic screen layout using an 80-column coding form to sketch your preliminary format.

4. Fill in the group header and format header options.

5. Transcribe the information from the screen layout to the form. First of all the fields must be numbered, usually in increments of 10. Then the field name, field length, spaces before, spaces after, and lines after options can be filled in directly from the screen layout.

6. Fill in the audit checks. Most of the audit checks can be derived from the file description and record layout, but it is possible to find other checks. In particular, the inserted constant, mandatory, crossfoot, and justify-in options may be used to provide additional checks. It is preferable to have ODESY find errors in the data as opposed to waiting until the application system detects them, so specify as many audit checks as possible.

7. Oeventh, obtain the output options from the file description and record layout.

8. Fill in the journal options. To obtain this information:

   a. Decide what is to be in the journal.
   b. Design the journal layout and write it down.
   c. Transcribe the information onto the form.

9. Execute the Format Maintenance program and enter the data as a new format.

SECTION 6

FORMAT MAINTENANCE PROGRAM

The Format Maintenance Program (FORMAINT) is an on-line interactive program which enables you to create screen formats and make changes to existing formats in the Formats file from an ODESY station. Section 5 describes the format options available and this section shows how those options may be manipulated.

Format data is entered under the control of a special group of screen formats contained in the program. The formats for format maintenance are shown in the second part of Appendix B. These special formats enable you to create a complete format in the workarea, starting with the group header and format header, and continuing with the individual fields and constants. The format data is checked for errors as it is transmitted, in the same way data is checked in the Edit Program. When the complete format has been entered, it is stored in the Formats file and you may begin work on another format.

FORMAINT allows existing formats to be amended or edited for new applications. This facility gives you great flexibility because, unlike a printed form, a screen format can be changed quickly whenever necessary. Also, you may display the format as it would appear for data entry before storing it in the Formats file, allowing you to make minor changes to the alignment of fields on the screen.

For reasons of security, no more than one person may enter format data at a time. You may execute FORMAINT while data is being entered or extracted; however, there are restrictions on how the Formats file may be accessed if these programs are executing.

Format maintenance is controlled by a set of commands which have been carefully designed to make use of the system as easy as possible and are similar to the data entry commands. These commands are discussed throughout the text and it may be useful to refer to Appendix A, which contains the syntax and semantics of the commands.

Appendix D contains a list of error and information messages pertaining to the Format Maintenance Program. An explanation is included with the number and standard text of each message.

FORMATS FILE AND FORMAT WORKAREA

Format maintenance uses both the Formats file and the format workarea. Formats are stored for use by other ODESY programs in the Formats file. The file contains a directory of group names. The maximum number of groups is 196, but there is no limit on the number of formats in each group. The total number of formats in the Formats file is limited only by the size of the file, which is 12000 records by default.

The format workarea is a temporary storage area. It enables you to change a format without affecting the permanent copy in the Formats file. The format workarea can contain one complete format at a time, consisting of a group header, a format header, and all the fields and constants of that format. The number of fields and constants in the workarea is limited by the sizes of the field and constant tables (refer to Section 4).

The first step in format maintenance, as in data entry, is to log in using the HELLO command with your operator number and password.

CREATING A FORMAT

A new format is created in the workarea.

ENTERING FORMAT DATA

Use the INSERT command to enter a new group header, format header, or field. ODESY progresses automatically from group header to format header to first, second, and all following fields, and as each is entered it is placed in the workarea.

Group Header Data

If the command parameter is omitted, ODESY responds by displaying the first group header format, Group Header 1. At this point, enter all of the group header format data for the first format and press the transmit key (XMT).

ODESY does not check the data of a format until it has been transmitted. If it finds any errors, the format is redisplayed with question marks in the fields in error. Now you must re-enter the erroneous fields and re-transmit all of the data. It is not possible to complete the entry of a group header, format header, or field which contains errors. If the error is not immediately obvious, it may be due to an error elsewhere; for instance, a range check on a non-numeric field. Appendix B and Section 5 contain advice on the relationships between format options.

If the data for format Group Header 1 has been entered correctly, ODESY automatically displays format Group Header 2 for the entry of the other group header options. Now enter this data.

Format Header Data

If there are no errors in the data for Group Header 2, ODESY displays Format Header for the entry of the format header data. Enter format header data in the same way you entered group header data. If there are no errors in the data for Format Header, the format header data is placed in the workarea and the first field format, Field Details 1, is automatically displayed.

Field Data

There are three main field formats: Field Details 1, Field Details 2, and Field Details 3. Field Details 1 contains mainly display options, Field Details 2 contains audit check options, and Field Details 3 contains output and journal options. The formats normally appear in consecutive order, with the next field beginning with Field Details 1. However, the order can be changed when certain field options are selected. For instance, if the field length (Field Details 1) is set to zero for a name-only field, Field Details 2 is omitted since there can be no audit checks.

Similarly, if the format type is set to D (display format), Field Details 3 is omitted because no data can be entered with this type of format. If an inserted constant, range check constant, or value check constant is selected (Field Details 2), one of the constant formats follows Field Details 3.

Inserted, Range, and Value Constants

The Inserted Constant format allows the entry of one inserted constant and is followed by Field Details 1 for the next field. Similarly, Range Check Constants allows the entry of three range check constants (default value, lower limit, and upper limit) and is followed by Field Details 1.

However, Value Check Constants, which allow the entry of a maximum of five value check constants, may be followed by either Value Check Constants again or by Field Details 1, depending on the number of constants entered. If five constants are entered, Value Check Constants appear again for the entry of further constants, but if four or less are entered this is considered to be the end of the constants and Field Details 1 is displayed. If exactly five constants are to be entered, leave Value Check Constants blank the second time, to indicate the end of the constants.

When all of the fields of the format have been entered, the automatic sequence of formats from Field Details 1 to Field Details 2 to Field Details 3 and back to Field Details 1 may be broken by entering another command. As with data entry, ODESY distinguishes it as a command by the slash character (/).

A field is not placed in the workarea until all of the format options for that field have been entered correctly. ODESY indicates this by displaying Field Details 1 again for the next field. For instance, assume you are entering a field with a range check and a set of invalid range check constants are entered. ODESY again displays format Range Check Constants, indicating the errors. If you ignore the errors and enter another command, all data for the current field is lost and must be re-entered, beginning with Field Details 1.

DISPLAYING OR LISTING A FORMAT

If you wish to look at the format before saving it, you can display the format in the workarea on the screen as it would appear for data entry.

The DISPLAY command has no parameters. It always displays the current contents of the workarea as the format would appear for data entry. Output-only fields are not shown. This command is useful for inspecting the alignment of fields and field names on the screen. Changes to alignment can be made with the AMEND command.

Use the LIST command to inspect the individual format options of the data in the workarea. The syntax of the command is similar to that of the INSERT command, and ODESY uses the Group Header Data formats for presenting the data. You may list the workarea, beginning with the group header, format header, a particular field, or a specific field details screen. Request the next "page" of the listing by pressing the transmit key, unless a single field details screen was requested. In all other cases, the group header automatically progresses to the format header, the first field, the second field, and so on. Break the sequence by entering another command.

SAVING A FORMAT

A format is not available for data entry until it is placed in the Formats file. Use the SAVE command to transfer the current contents of the workarea to the Formats file, and to update the group and format directories in the file. The command has no parameters.

ODESY keeps a record of what is in the workarea and what has been changed. For example, if the format header or one of the fields has been amended, the format is "present and changed," but the group header is "present and unchanged." So when the SAVE command is used, the format is updated but the group header is not. Similarly, if the operator attempts to use the INSERT command to enter a new group or format while there is changed data in the workarea, ODESY asks the operator to save or delete the format.

The SAVE command makes permanent changes to the Formats file. You cannot do a SAVE if the Extraction program is running. You can use SAVE during data entry if the On-Line Format Maintenance option is included in the system, but only if there are no batches open in the group being changed. If a group is being used by EDIT, you can save the format under a temporary test group name.

If the On-Line Format Maintenance option is excluded, or the Extraction program is running, you must temporarily store unsaved data by dumping it from the workarea to a Format Dump file. When data entry or extraction is complete, the Dump file can be loaded to the Formats file.

CHANGING A FORMAT

Use the GET command to transfer a group header or a group header and format from the Formats file to the workarea. The GET command has two parameters, indicating the group name and format name. If the format name is omitted, only the group header is placed in the workarea.

Various editing facilities enable you to manipulate the data in the format workarea. You may make changes to existing fields in the format (amendments), add missing fields to the format (insertions), and delete unwanted fields from the format (deletions).

Other editing facilities enable you to reorganize the screen layout by rearranging the order of the fields. Additionally, the COPY command allows a new format to be built up in the workarea by duplicating fields from an existing format.

AMENDING FIELDS USING ALL SCREENS

Use the AMEND command to change individual format options of the data in the workarea. It has the same syntax as the LIST command, and ODESY uses the same special formats (Group Header 1, Group Header 2, etc.) for presenting the data.

If you enter the AMEND command followed by a field sequence number, ODESY responds by displaying the format Field Details 1, containing the data for the field requested. You

may now change the field options displayed. Indicate completion by pressing the CLEAR HOME key and transmitting. If there are no changes, just press the transmit key. In either case, the next format, Field Details 2, is displayed with further field options, which you may change.

The AMEND command progresses automatically from one field to the next. The sequence is terminated by entering a new command. Be careful when amending a field that all of the format options for that field have been properly processed before entering a new command, because the amended field is not added to the workarea until ODESY has a complete set of format options for the field. ODESY indicates it has finished by displaying Field Details 1 again for the next field.

In general, you amend a format header and group header in the same way, but a group header does not automatically progress to the format header.

AMENDING FIELDS ONE SCREEN AT A TIME

To make a change involving only one FIELD DETAILS screen, specify the screen number following the field sequence number in the AMEND command. For example, to change the output position of a field, enter:

```
/AMEND <field sequence #> 3
```

ODESY will display the Field Details 3 screen. Transmitting that screen with the changed data will amend the field.

If FIELD DETAILS 2 is specified in this manner and the constant, range, or value check feature is used, all constant screens must be transmitted before the AMEND will go into effect.

DELETING FIELDS

Use the DELETE command to delete unwanted data from the format workarea. You may delete either a single field or the entire workarea. You must delete unwanted data from the workarea before further format maintenance can continue.

RESEQUENCING FIELDS

Use the RESEQUENCE command to reallocate sequence numbers to all of the fields of the format in the workarea. Specify the first new sequence number and the new increment. After resequencing, the fields remain in the same order, but are separated uniformly. This command is useful when a format has been reorganized or when there are no spare sequence numbers.

MOVING FIELDS

Use the MOVE command to transfer a particular field or range of fields to a new position on the screen by changing the sequence numbers. The screen layout is determined by the order of sequence numbers in the fields, so moving a field alters its position on the screen.

DUPLICATING FIELDS

Use the COPY command to duplicate a field or range of fields and place the copy at a new position in the format. The COPY command is particularly useful when a format is to produce multiple records in the application file. Some fields in the format will be identical; enter such fields once and then use the COPY command.

The COPY command can also be used to copy fields from existing formats in the Formats file. A set of fields which are common to several formats can be created once and copied. An extension of this is the case in which several groups have a format in common, for instance a batch header format. You need only to create the format once and copy all the fields for the other groups.

The MOVE command and the COPY command are used to reorganize the screen layout of a format. The difference between them is that a copy produces two fields, an original and a duplicate, and a move produces only one field, moved to a different location.

ADDING A FORMAT TO A GROUP

The INSERT command may also be used to add a new format to an existing group. To do this, obtain the group header from the Formats file using the GET command and then the INSERT command with the FORMAT parameter to begin entry of the format header.

A new field may be inserted into an existing format in a similar manner. Obtain the group header and format from the Formats file using the GET command. Then enter the INSERT command followed by a field sequence number.

REMOVING A FORMAT

Use the REMOVE command to remove a group or format from the Formats file. It has two parameters, indicating the group name and format name. If the format name is included, only that format is removed. If the format name is omitted, the entire group is removed from the file. The REMOVE command cannot be used if the Extraction program is executing, or if the group affected is in use by EDIT. See the system configuration parameter On-Line Format Maintenance in Appendix B.

The REMOVE command permanently removes the format or group of formats from the formats file. Before removing a format, a backup copy of it should be made. The Extraction program cannot extract or delete a record from the tank file unless its format is present in the formats file.

The last step in format maintenance, as in data entry, is to log out using the BYE command.

PRINTING A FORMAT

A Format Maintenance Report shows details of a format and its screen layout. Using the PRINT command, you may request a printed report of a single format, all the formats in a particular group, the entire Formats file, or the format in the workarea.

The report for a specific format consists of six parts:

1. Group header options.
2. Format header options.
3. Screen layout.
4. Display options.
5. Audit check, output, and journal options.
6. Summary totals.

Group header options, format header options, display options, and audit check, output, and journal options correspond with the input formats in Appendix B. The screen layout is a representation of how the format appears during data entry. The size of the screen may affect the appearance of the format. The screen size used in the report, either 80 X 24, 80 X 12, 40 X 12, or 32 X 8, matches the screen size of the ODESY station requesting the report. The summary totals indicate how many fields and constants are used by the format. They are used to compute the memory requirements of the Edit and Extraction Programs.

Use the PRINT DIRECTORY command (PD) to request a displayed directory of all format names in a particular group or of all group names in the Formats file. This command is helpful in checking that a particular format or group is in the Formats file.

Figure 2 shows a sample Format Maintenance Report.

```
B1000 ON-LINE DATA ENTRY SYSTEM        F O R M A T   M A I N T E N A N C E   R E P O R T  TIME 11:26  DATE 01/23/81  PAGE   1
   GROUP NAME  SYS                                     FILE NAME            UDESYSETUP/INPUT    DATE LAST CHANGED 01/23/81

   RECORD SIZE  80  BLOCKED  9   RECORDS/AREA  450   AREAS 40   ON DISK                       REPORT  DATE FORMAT  AS SYS
   GROUP SIGN CONVENTION       MSD

   FORMAT NAME SYS STA            NEXT FORMAT SYS STA    SECURITY  50       VERIFY           DATE LAST CHANGED 01/23/81
                                  LINE ADVANCE 0         CURSOR ADVANCE  0

                  ****************************************************************************
                  *  00G 0000 STA          [ ][STA ]                                         *
                  *MAINTENANCE ACTION:I,D        [I]                                         *
                  *LOGICAL STATION NUMBER       [  ]                                         *
                  *STATION NAME           [          ]                                       *
                  *STATION BUFFER SIZE        [1920]                                         *
                  *FORMS SET:A,B,C               [C]                                         *
                  *TERMINAL TYPE                [46]                                         *
                  *CONTROL STATUS: ,C            [ ]                                         *
                  *                                                                          *
                  *                                                                          *
                  *                                                                          *
                  *                                                                          *
                  *                                                                          *
                  *                                                                          *
                  *                                                                          *
                  *                                                                          *
                  *                                                                          *
                  *                                                                          *
                  *                                                                          *
                  *                                                                          *
                  *                                                                          *
                  *                                                                          *
                  *                                                                          *
                  ****************************************************************************

DISPLAY SPECIFICATIONS
                                                SPACE ADVANCE  LINE       HIGHLIGHT  DUP  DISP
SEQ        NAME                         LENGTH  BEFORE/AFTER  ADVANCE                IN   VER

 10                                        3       0     1       0
 20   MAINTENANCE ACTION:I,D               1       0     6       1
 30   LOGICAL STATION NUMBER               2       0     5       1
 40   STATION NAME                        10       0     7       1
 50   STATION BUFFER SIZE                  4       0     6       1
 60   FORMS SET:A,B,C                      1       0    13       1
 70   TERMINAL TYPE                        2       0    14       1
 80   CONTROL STATUS: ,C                   1       0    10       0

B1000 ON-LINE DATA ENTRY SYSTEM        F O R M A T   M A I N T E N A N C E   R E P O R T  TIME 11:26  DATE 01/23/81  PAGE   2


OTHER SPECIFICATIONS
SEQ  TYPE  CNS/VAL  MAND  DATE  CDV  CROSSFOOT   BATCH   JUST  OUT   WRITE  DUP  JUST  PACK  BLANK  SIGN  NAME   DATA  PRINT
           /RANGE                                TOTALS   IN   POSN  AFTER  OUT   OUT   OUT  FILL          COL    COL   AFTER

 10   U      C                                                   1                                                  1
                    CONSTANTS    STA
 20   AN     V                                                   4                                                  5
                    M
                    CONSTANTS    I D
 30   UN     R                                                   5                  R                        10    40
                    M
                    CONSTANTS       01 99
 40   AN                                                         7                  L                        47    69
 50   UN     R                                                  17                  R                        84   112     C
                    CONSTANTS    1920 0010 4095
 60   AN     V                                                  21                                           10     4
                    M
                    CONSTANTS    C A B
 70   UN     R                                                  23                  R                        47     7
                    M
                    CONSTANTS    46 32 66
 80   AN     V                                                  22    C                                      84    11     C
                    CONSTANTS      C

**** TOTALS  ****          FIELDS INPUT         8          FIELDS OUTPUT                    8
                           INPUT DATA SIZE     21          CHARACTERS OF CONSTANT DATA     34
                           OUTPUT DATA SIZE    24          CONSTANTS REQUIRED BY EXTRACT    3
                           FIELDS PRINTED      14          MINIMUM SCREEN SIZE OF FORMAT 8 BY 32
```

Figure 2. Format Maintenance Report

FORMAT DUMP FILES

A Format Dump file is a serial file containing format data. It is normally used as a temporary or intermediate file. It is very different from the Formats file. It has three main uses:

1. Temporary storage for formats, for instance when the Formats file is in use by EXTRACT.
2. Format conversion aid.
3. Library file.

There can be many Format Dump files, but only one Formats file. A Dump file is accessed during format maintenance by the DUMP command and the LOAD command.

The DUMP command transfers format data from the Formats file or the format workarea to a Format Dump file. You can dump the entire Formats file, a specific format, or a group of formats. Refer to the Dump file by name. If the file already exists, format data is added to the end of the file. If it does not exist, a new file is created.

The DUMP command provides a solution to the problem of saving the format in the workarea while the Data Entry program or the Extraction program is running. Use the DUMP command to temporarily dump the workarea to the Dump file. The contents of the Dump file can be loaded to the Formats file with one command when Data Entry and Extraction are inactive.

Because new format data is added to the end of an existing Dump file, it is possible to have several copies of the same format in a single file. Although this wastes disk space, it should not result in errors. When ODESY loads the format data again, each copy of the format is loaded in turn, over-writing the previous copy. Only the last copy of the format remains in the Formats file after the LOAD command is complete.

The LOAD command transfers format data from a Format Dump file to the Formats file or the format workarea. You can load the entire Dump file, a specific format, or a group of

formats. When loading to the workarea, you may load only a single format. The LOAD command also allows the user to load a format or a group of formats under a new group name or format name.

The Formats file is a structured file of format data with directories and forward and backward pointers to the records of the file. A Format Dump file is a simple serial file of format data with no internal structure. Due to this simplicity of structure, the Dump file is used to carry format data from a Formats file used by a previous release of B 1000 ODESY (small systems ODESY) to the Dumpfile Conversion Program, provided with the latest small systems ODESY release, where it can be converted for use with the newer release. This may be necessitated by a change in the structure of the Formats file between the old and new releases. This eliminates the need to re-enter all of the format data.

Another use of the Dump file is to carry format data between small systems ODESY and medium systems ODESY. A medium systems ODESY format is different from a small systems format, so to use a format on both systems it is necessary to go through a conversion procedure. This is done by the Dump File Conversion Program supplied with the medium systems ODESY release and, as the name implies, it converts the format data in a Dump file. It is normally a once-only change to convert formats from another system. The medium systems ODESY Dump File Conversion Program converts a small systems Dump file to a medium systems Dump file.

A Format Dump file can also be used as a library file. It can store old, seldom-used, and alternative versions of formats. Also, if two different sites, each with its own computer, are executing ODESY, Dump files provide a means of carrying common formats between the two sites. The alternative is to copy the entire Formats file.

FORMAT MAINTENANCE EXAMPLE

This example illustrates the main features of format maintenance. The instruction "enter" is used to mean "type in and transmit."

#EX Odesy/FormAint

First, as in data entry, log in using the HELLO command.

Enter: /HELLO 1 SUP1

If your log-in entry is valid, ODESY replies:

```
START OF SESSION
```

Suppose you are to enter a new format, with group name INV and format name T1. First enter the group header details.

Enter: /INSERT

ODESY replies:

```
 INSERT GROUP                         P.1

     GROUP HEADER 1
GROUP.NAME                   [    ]
FILE.NAME                         [          ]
             [          ] / [          ]
RECORD.SIZE                       [   ]
BLOCKED                           [  ]
RECORDS.PER.AREA                  [    ]
NUMBER.OF.AREAS                   [  ]
MEDIUM: ,T                        [ ]
         .                                [ ]
```

The first line contains control information, such as the current command (INSERT) and the item being maintained (Group Header).

Enter the following group header data:

```
 INSERT GROUP                P.1

     GROUP HEADER 1
GROUP.NAME              [INV ]
FILE.NAME                    [          ]
            [INV01     ] / [          ]
RECORD.SIZE                  [80 ]
BLOCKED                      [9 ]
RECORDS.PER.AREA             [90  ]
NUMBER.OF.AREAS              [99]
MEDIUM: ,T                   [ ]
                                      [ ]
```

There are no errors and ODESY replies with the second group header format. Enter nothing here because the default values are sufficient. Note that ODESY has established that the group name is INV and has placed it in the control line.

```
 INSERT GROUP  INV           P.2

     GROUP HEADER 2

RECORDING.MODE: ,A,B                  [ ]
LABEL.TYPE: ,A,U                      [ ]
DATE.FORMAT: ,E,U,I,J                 [ ]
SIGN.CONVENTION: ,1,2,3               [ ]
JOURNAL: ,J                           [ ]
JOURNAL.INPUT.REC.NO                  [   ]
JOURNAL.OUTPUT.REC.NO                 [   ]
                                        [ ]
```

The group header is complete and ODESY now displays the format header format. Enter the following data:

```
 INSERT FORMAT INV           P.1

     FORMAT HEADER
FORMAT.NAME                        [T1  ]
NEXT.FORMAT                        [T1  ]
SECURITY.LEVEL                       [40]
CURSOR.ADVANCE                       [  ]
LINE.ADVANCE                         [2 ]
NO.VERIFY: ,N                         [ ]
FORMAT.TYPE: ,B,D,K,S,C,U             [ ]
USER.PROGRAM.NAME           [           ]
        [          ] / [              ] [ ]
```

D - no data entry ; display info to user
B - batch totals
K - sends data no checking on user program

Again, there are no errors and the first field format appears. ODESY offers a field sequence number of 10 as the default. Note that the control line contains the group name, format name, and field sequence number, and that the field sequence number is automatically placed in the field provided. It may be overridden if desired. Enter the following data:

```
 INSERT FIELD  INV T1   010 P.1

     FIELD DETAILS 1

SEQUENCE.NUMBER                   [010]
FLD.NAME [CUSTOMER                     ]
FIELD.LENGTH                     [8  ]
SPACES.BEFORE/AFTER.DESC          [4 ] [12]
LINES.AFTER.DESCRIPTION           [1 ]
HIGHLIGHT: ,B,U,R,S               [R]
DUPLICATE.IN: ,D                  [ ]
VERIFY.METHOD: ,D,N,I             [D]    [ ]
```

24 rows
80 column

B - bright video
U - underline video
R - reverse video
S - secure video

D - default refer, data key most recent the error
N - no verify
I - data keyed initially is displayed

The first field format contains the display options. This is followed by the second field format which contains the audit check options. Enter the following data:

```
 INSERT FIELD  INV T1   010 P.2

     FIELD DETAILS 2

FIELD.TYPE: ,N,S,O,T                          [N]
CONSTANT/RANGE.CK/VALUE.CK: ,C,R,V            [ ]
MANDATORY: ,M                                 [M]
DATE.CHECK: ,D                                [ ]
CDV: ,1-9; XFOOT: ,+,-              [3]       [ ]
BATCH.TOTAL: ( ,-,I,N)( ,1-10)      [ ]       [ ]
JUSTIFY.IN: ,L,R                              [ ]
                                              [ ]
```

The third field format follows the second. It contains output options and display options. Enter the following data:

```
 INSERT FIELD  INV T1   010 P.3

     FIELD DETAILS 3

OUTPUT.POSITION                     [4   ]
WRITE.AFTER: ,W,S,C                 [ ]
DUPLICATE.OUT: ,D,C                 [D]
JUSTIFY.OUT: ,L,R; PACKED: ,P,H     [R]       [P]
BLANK.FILL: ,B; SIGN: ,+,-          [ ]       [ ]
NAME.COLUMN                         [   ]
DATA.COLUMN                         [   ]
PRINT.AFTER: ,P,S,C                   [ ]     [ ]
```

If an inserted constant, range check constant, or value check constant is selected, the third field format is followed by a constant format. For example, enter the following data for field 60:

```
  INSERT V-CONS INV T1  060 P.4

[       ]
[NUTS   ]
[BOLTS  ]
[SCREWS]
[       ]


                                    [ ]
```

When all of the fields have been entered, you can view the format as it would appear for data entry.

Enter: /DISPLAY

ODESY replies:

```
DISPLAY          INV T1          P.1

  CUSTOMER              [              ]
  ACCT NUMBER           [         ]
  PART NUMBER      [                   ]
  DESRIPTION       [                   ]
  QTY  [   ]       UNIT  [             ]

       [           ]
```

If all of the format data is correct, save the contents of the workarea in the Formats file.

Enter: /SAVE

ODESY replies:

```
INV T1 SAVED
```

The contents of the workarea are not changed by the SAVE command. If further maintenance is required for this format, you may do it without reloading it to the workarea. However, if you want to work on other formats, you must use the GET command.

At this point, you may list or amend format data in the workarea. For instance, you may inspect the format header.

Enter: /LIST FORMAT

ODESY replies:

```
 LIST   FORMAT INV T1        P.1

     FORMAT HEADER
FORMAT.NAME                    [T1  ]
NEXT.FORMAT                    [T1  ]
SECURITY.LEVEL                   [40]
CURSOR.ADVANCE                   [00]
LINE.ADVANCE                     [02]
NO.VERIFY: ,N                     [ ]
FORMAT.TYPE: ,B,D,K,S,C,U         [ ]
USER.PROGRAM.NAME       [           ]
        [           ] / [           ] [ ]
```

Or you may make changes to field 70.

Enter: /AMEND 70

ODESY replies:

```
 AMEND  FIELD  INV T1   070 P.1

     FIELD DETAILS 1

SEQUENCE.NUMBER                 [070]
FLD.NAME [                             ]
FIELD.LENGTH                   [008]
SPACES.BEFORE/AFTER.DESC        [09] [00]
LINES.AFTER.DESCRIPTION         [01]
HIGHLIGHT: ,B,U,R,S             [ ]
DUPLICATE.IN: ,D                [ ]
VERIFY.METHOD: ,D,N,I           [ ]   [ ]
```

Change the data as required on the FIELD DETAILS 1 screen. Transmit that screen and ODESY will display the rest of the screens associated with the specified field, followed automatically by the next field in the format. All of the screens for the current field must be transmitted before the AMEND will go into effect.

As an alternative, if only some information (for example, the information on Field Details 1) needs to be changed, specify the field details screen you want following the field sequence number such as:

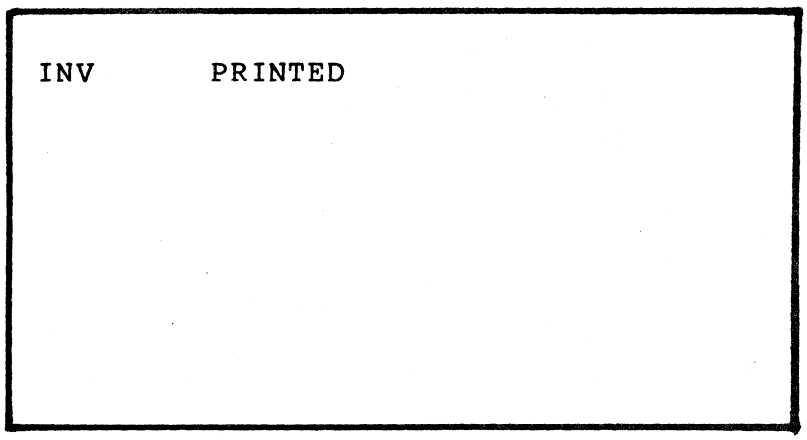
```
/AMEND 70 1
```

ODESY replies to this command with the same screen (above). In this case, however, only the FIELD DETAILS 1 screen will be displayed. The AMEND will be done when this screen is transmitted. Similarly, the FIELD DETAILS 2 or 3 screens can be requested individually for changes that affect only one screen of the field specifications.

To obtain a printed report, use the PRINT command. Suppose you wish to print all of the formats in group INV.

Enter: /PRINT INV

ODESY replies briefly with the statement PRINT STARTED and then overlays the line with:



Finally, log out.

Enter: /BYE

ODESY replies:

```
END OF SESSION
```

SECTION 7

EDIT AND EXTRACTION PROGRAMS

ODESY is designed primarily for the entry of batches of data that are later processed by an application system. Auditing facilities enable the terminal operator to check data for errors during the entry and verification stages so that the application system may process error-free data. This procedure is completed by the terminal operator through the use of the Edit Program. Refer to the B 1000 Series On-Line Data Entry System (ODESY) Terminal Operator's Manual, form 1131851, for complete instructions in the entry and verification of data.

EDIT PROGRAM

The Edit Program (EDIT) is an on-line interactive program capable of coordinating the entry of data from a large number of ODESY stations. This program is used to create batches of error-free data in the Tank file by allowing each terminal operator at an ODESY station to enter a batch of data, one record at a time, under the control of screen formats and a set of commands. The data is checked for errors and stored in the Tank file when it is correct. If there are errors in the data, the program informs the operator who must correct the errors.

Once data has been entered, it may be recalled by the operator. There is an optional verification stage in which the operator re-enters the data. It is compared with the original entry and differences are reported to the operator. ODESY also enables the operator to make minor changes to the data, or insert or delete records. The operator indicates that a batch is complete by scheduling it for extraction. The batch in the Tank file may then be processed by the next ODESY program.

The Edit Program obtains information such as details of operators, stations, and check digit verification schemes from the System file. Each operator must log on, using a number and password, and this is checked against the file. Similarly, each station is checked against the file. Both the station and the operator must be valid.

If the Data Entry Program must be terminated from the ODT because no valid stations or operator passwords are available, enter:

<mix no>AXTER NOW

The formats are obtained from the Formats file. They indicate what is to appear on the screen and which audit checks are to be applied.

EXTRACTION PROGRAM

The Extraction program (EXTRACT) takes batches of data from the Tank file and creates serial files on disk or tape suitable for processing by an application system. EXTRACT also deletes from the tank file the batches that are no longer needed.

Batches are marked for extraction or deletion through the Edit program by using the SCHEDULE command alone or with the DEL option. To "unschedule" a batch for extraction or deletion, simply open it.

The Batch Status Print program can be executed before each Extraction to ensure that the correct batches are scheduled for extraction or deletion.

For information about how EXTRACT may create secured files, see Section 10 under "Usercode Security."

RUNNING EXTRACT AND EDIT TOGETHER

Extraction can be performed alone or at the same time that data is being entered. However, it is important to be consistent with the method of extraction that you use within a given extraction run. In other words, do not initiate or terminate EDIT while EXTRACT is executing. This is because when EDIT is in the mix, EXTRACT does not write directly to the ODESYDATA files under some circumstances. In these cases, EDIT updates the data files to avoid contention problems.

For example, when EDIT is running, EXTRACT stores operator statistics in a temporary file called the OPSTATS file. After all batches are processed, EXTRACT asks EDIT (via queue file communication) to transfer this data into the SYSTEM file. When EDIT is not running, EXTRACT writes directly to the ODESYDATA files, and no queue file communication is involved.

HOW EXTRACT WORKS

During a run of the Extraction program, each of the 999 possible batches is scanned in order by batch number, and any batch scheduled for extraction or deletion is processed. All such batches will be listed in an Extraction report with a 1-line summary of the action taken.

Individual records are extracted field by field based upon the format specifications. An output (application) file record is written each time EXTRACT encounters a write specified in a field's "WRITE - AFTER" option (provided the record is not blank). For more information about the Extraction process on a field level, see Section 5 under "Write Options."

When the Extraction program finds a batch that is scheduled for extraction, it uses the group name to obtain details for the file to be created. If EXTRACT is to create a magnetic tape file, a new tape file is opened, and extracted records are written to the tape. If EXTRACT is to create a disk file, it checks to see if such a file already exists. If it does, EXTRACT will add records to the end of it. Otherwise, EXTRACT will create a new disk file according to the specifications in the group header. If this file becomes full, the message "OUTPUT FILE FULL" is printed on the Extraction report, and the records from the current batch that were added to the file are removed.

NOTE

EXTRACT cannot add records to a CANDE file or to any file previously closed with CRUNCH, because the file is considered full.

THE EXTRACTION REPORT

The Extraction report shows the following information by batch:

1. The operator number of the operator who scheduled the batch.

2. The group name, output file name, and the output file medium (if tape).

3. The number of records entered, verified, and re-extracted. (A record is counted as being re-extracted if the record has been previously extracted, and the data has not been modified in the meantime.)

4. The number of records in the output file both before and after the release of a batch.

5. The number of lines printed (if any) in an Extraction Journal.

6. Information regarding the outcome of the extraction process for that batch, such as "batch extracted."

In addition to creating the Extraction Report, the Extraction program also gathers statistical information for the Statistics Analysis program.

JOURNAL FACILITIES

A user-defined Extraction Journal may be created for a hard-copy record of extracted data. To do this, the report option must be set for the group.

The contents of the journal depend upon the values of the Print Column, Print After, and Print Name options for each field of each format used. Printing data on the journal is specified independently from writing a record to the output file being created. Thus, secure or unimportant data need not be printed on the journal. For example, the journal may serve as an exact audit trail of what is extracted, or it may list only key fields from certain formats.

DELETION

A batch is not automatically deleted after extraction. If no errors are found by the application system, the batch should be explicitly deleted from the Tank file with the SCHEDULE DELETE command.

When the Edit program receives the command to delete a batch, the only action taken is to mark the batch for deletion. The actual data in the batch is not deleted until the next run of the Extraction program. At any point up until that time, the batch may be "unscheduled" by opening it, and the batch will not be deleted.

If you do not wish to retain extracted batches, then SCHEDULE and SCHEDULE DELETE may be used to force the deletion of a batch as soon as the application file has been created. You also may use SCHEDULE DELETE only, since a batch will not be deleted until it has been extracted at least once.

If the tank file becomes full, run the Edit program to schedule a few unwanted batches for deletion, and then run the Extraction program to return the data records to the chain of free records. If the Tank file regularly becomes full, it may be advisable to increase the limits of the file. See Section 3.

JOURNAL-ONLY AND NO-JOURNAL FACILITIES

The Journal-Only facility lets you do a trial extraction run to ensure that the correct data will be extracted and the correct batches are scheduled for extraction and deletion.

The Journal-Only facility is selected by setting external switch 2 to 1. All that is produced from this type of execution is an Extraction Journal and an Extraction Report. At the top of both reports appear the words JOURNALS ONLY. No batches are actually extracted or deleted.

To select the No-Journal facility, set switch 2 to 2. No journal will be produced from the extraction run, but in all other respects it will be a normal run involving extraction, deletion, and the creation of an Extraction Report.

```
B1000 ON-LINE DATA ENTRY SYSTEM          E X T R A C T  R E P O R T          TIME 11:54  DATE 01/28/81  PAGE   1

            LAST REPORT:   15   DATED: 01/20/81                                              REPORT:  16

BATCH  SCHD  GROUP  OUTPUT   (PACK)        FILE NAME          RECORDS IN BATCH      OUTPUT RECORDS   LINES
        BY   NAME   MEDIUM                                ENTERED VERIFIED RE-EXTD  BEFORE   AFTER  PRINTED

 999     1   SYS                        ODESYSETUP/INPUT       92        0      92       5      162      191 BATCH EXTRACTED
```

Figure 3. Extraction Report and Extract Journal

```
B1000 ON-LINE DATA ENTRY SYSTEM          E X T R A C T I O N  J O U R N A L          TIME 11:53  DATE 01/28/81  PAGE   1

   BATCH 999              DISK FILE              ODESYSETUP/INPUT

CONI     MAX FORMATS/FIELDS              0100   MAX CHARS OF CONST DATA     0200   MAX USER PROGRAMS              00
         MAX IN SZ/ZINK FULL    0160      95   REC SZ OF TANK FILE:A,B,C        A   PAGE SZ/DATE TYP:U,I,E      60 U
0630     MAX FIELDS - FORMAINT            063   MAX CHARS OF CONST DATA     0500   MAX CHARS SCRATCHPAD CNS    0100
         MAX FORMATS - EXTRACT             10   MAX FIELDS                  0200   MAX CHARS OF CONST DATA     1000
         MAX OUTPUT RECORD SIZE           400
NNNN     DUPLICATE ANY: Y,N                N   RESTRICT OPTIONS: R,N              ACCEPT          N CONTROL       N
         DELETE      N INSERT              N   SCHEDULE        N SCHD.DEL         N EXT.SCHD      N TERMINATE     N
         INCLUDE OPTIONS/COMMANDS:I,E
         OPTS-COV      I BTOT              I   XFOOT           I
         CMDS-ACCEPT   I CONTROL           I   DELETE          I DISPLAY       I   INSERT          I READY         I
         EXT.SCH       I SS                I   STATUS          I WRU           I   DS              I
I01 I    LOGICAL STATION NUMBER           01   STATION NAME            10802      STATION BUFFER SIZE        1920
         FORMS SET:A,B,C                   A   TERMINAL TYPE              42      CONTROL STATUS: ,C              C
I01 I    OPERATOR NUMBER                  01   OPERATOR NAM    SUPERVISOR 1       OPERATOR SECURITY LEVEL     99
         OPERATOR TYPE:O,S                 S
I02 I    OPERATOR NUMBER                  02   OPERATOR NAM    SUPERVISOR 2       OPERATOR SECURITY LEVEL     99
         OPERATOR TYPE:O,S                 S
I03 I    OPERATOR NUMBER                  03   OPERATOR NAM    SUPERVISOR 3       OPERATOR SECURITY LEVEL     75
         OPERATOR TYPE:O,S                 S
I04 I    OPERATOR NUMBER                  04   OPERATOR NAM    SUPERVISOR 4       OPERATOR SECURITY LEVEL     50
         OPERATOR TYPE:O,S                 S
I05 I    OPERATOR NUMBER                  05   OPERATOR NAM    SUPERVISOR 5       OPERATOR SECURITY LEVEL     25
         OPERATOR TYPE:O,S                 S
I06 I    OPERATOR NUMBER                  06   OPERATOR NAM    OPERATOR 10        OPERATOR SECURITY LEVEL     90
         OPERATOR TYPE:O,S                 O
I11 I    OPERATOR NUMBER                  11   OPERATOR NAM    OPERATOR 11        OPERATOR SECURITY LEVEL     80
         OPERATOR TYPE:O,S                 O
I12 I    OPERATOR NUMBER                  12   OPERATOR NAM    OPERATOR 12        OPERATOR SECURITY LEVEL     60
         OPERATOR TYPE:O,S                 O
I13 I    OPERATOR NUMBER                  13   OPERATOR NAM    OPERATOR 13        OPERATOR SECURITY LEVEL     40
         OPERATOR TYPE:O,S                 O
I14 I    OPERATOR NUMBER                  14   OPERATOR NAM    OPERATOR 14        OPERATOR SECURITY LEVEL     20
         OPERATOR TYPE:O,S                 O
I15 I    OPERATOR NUMBER                  15   OPERATOR NAM    OPERATOR 15        OPERATOR SECURITY LEVEL     10
         OPERATOR TYPE:O,S                 O
001S     001 SECURITY LEVEL ERROR
001S     002 CLOSING DOWN-END YOUR WORK NOW
003C     003 CLOSE FORCED-YOU ARE LOGGED OUT
003C     004 SYSTEM ERROR-LAST COMMAND LOST
005T     005 THE BATCH IS NOW CLOSED
005T     006 THE RECORD IS ACCEPTED
007N     007 NOW SCHEDULED FOR EXTRACTION
007N     008 RECOVERY STARTED
009A     009 A BATCH IS ALREADY OPEN
009A     010 YOU ARE NOT ENTERING DATA
011A     011 AN OPERATOR IS LOGGED IN
011A     012 NO OPERATOR IS LOGGED IN
013Y     013 YOU ARE LOGGED IN ELSEWHERE
013Y     014 YOUR NAME ISN'T ON FILE
```

SECTION 8

AUXILIARY PROGRAMS

There are five auxiliary ODESY programs: the Batch Status Print program, used in data control; the Statistics Analysis program, a management aid to effective system use; the Tank File Conversion program, a conversion aid; and the Dumpfile Conversion program, also a conversion aid. The Tank File Analysis program is also a debugging aid which may be used by your Burroughs System Support Representative.

BATCH STATUS PRINT PROGRAM

The Batch Status Print program (PRINT) produces a report showing all existing batches and their current status. It can be used for data control to determine how many batches have been entered and by whom, and which ones are scheduled for extraction or deletion.

For each batch, the report shows the operator who first opened the batch, the date it was first opened, the date it was last accessed, the operator who scheduled the batch for extraction (if any), the number of records in the batch, the group associated with the batch, and the values of four status flags. These flags are enter status, verify status, extract status, and delete status. The information produced by this report is similar to that produced by the Edit program STATUS command.

To ensure that the correct batches are scheduled for extraction or deletion, execute this program before executing the Extraction program.

Figure 4 shows a sample Batch Status Print report.

B1000 ON-LINE DATA ENTRY SYSTEM B A T C H S T A T U S P R I N T R E P O R T TIME 11:38 DATE 01/28/81 PAGE 1

BATCH	NUMBER OF RECORDS	GROUP	OPEN OPR	SCH OPR	STATUS FLAGS ENTER	VERIFY	EXTRACT	DELETE	DATE CREATED	DATE LAST ACCESSED	IN USE AT
3	4	T4	1	1	COMPLETED		DONE		01/17/81	01/17/81	
4	3	T1	1	1	COMPLETED		DONE		01/19/81	01/19/81	
5	5	HOPE	1	1	COMPLETED		DONE		01/19/81	01/20/81	
999	92	SYS	1	1	COMPLETED		DONE		11/24/80	11/24/80	

Figure 4. Batch Status Print Report

STATISTICS ANALYSIS PROGRAM

The Statistics Analysis program (STATS) produces two statistics reports: the Formats report and the Operator report. Using the data gathered by the Edit and Extraction Programs, each report summarizes the data that has been extracted since the last execution of the Statistics Analysis program. These reports may be used by management to identify sources of inefficiency in the system, and as a general indication of system use.

An identifying transaction number is printed at the top of each report. Each time the program is executed, this number is increased by one. The report also shows the transaction number of the last report and the date on which it was produced.

The Formats report shows the use of each format and the volume of data for each application. It includes the number of records extracted for each format and group of formats.

The Operator report shows the number of extracted records entered and verified, the approximate number of keystrokes for Enter mode and Verify mode, the elapsed time (the time spent logged in at a terminal), the number of records per hour, and the approximate number of keystrokes per hour. The report includes a detail line for each operator and a total for all operators.

The Operator report was designed to serve as a general guide to system use and not as an analysis of operator performatnce. The method of gather Operator statistics should be understood to accurately interpret the Operator report. Operator statistics are gathered by the following programs:

1. EDIT. This program notes the time each operator logs on and off, and stores this information in the approrpiate operator record in the system file.

2. EXTRACT. This program looks at the operator numbers of both the entering and verifying operator stored in each data record as it extracts the batch. It bumps those operators' record totals by one for each record extracted, and their

character totals by the length of the field scanned for each field encountered. It does not know if the field was an inserted constant, was supplied by a user program, or was entered by the operator. It also does not know whether the field was displayed or rekeyed for verification. Thus, it does not provide an exact account of the number of characters actually keyed by an operator.

Each time STATS is run, the time, records keyed, and records verified fields for each operator are reset to zero in the system file. To optimize the accuracy of the reported statistics, all batches entered since the previous run of STATS should be extracted before reexecuting the STATS program. If all of the data entered or verified since the last STATS run is not extracted before running STATS, the report can become even less accurate. For example, it can show no records entered but much time for an operator, or vice-versa.

For best results, execute STATS once per week or once per month, and consider the reported values to be only an approximation of actual activity.

Figure 5 shows a sample Formats report and figure 6 shows a sample Operator report.

```
B1000 ON-LINE DATA ENTRY SYSTEM   S T A T I S T I C S   A N A L Y S I S   R E P O R T  TIME 11:04  DATE 01/23/81  PAGE  1
                                                                                                          REPORT#  1

                                         F O R M A T S   R E P O R T
  GROUP SYS
                 FORMAT     RECORDS ENTERED
                 CONF            0
                 CON2            0
                 CN3R            0
                 CN3I            0
                 STA             0
                 OPR             0
                 TEXT            0
                 CMD             0
                 CMD2            0
                 CMD3            0
                 CMD4            0
                 CDV             0
                 PRNT            0
                 SQUA            0
                 CON3            0

  TOTAL FOR GROUP                0
```

Figure 5. Formats Report

BLOOD ON-LINE DATA ENTRY SYSTEM S T A T I S T I C S A N A L Y S I S R E P O R T TIME 11:04 DATE 01/23/81 PAGE 1

REPORT# 1

O P E R A T O R R E P O R T

OPERATOR	NUMBER OF RECORDS ENTERED / VERIFIED		NUMBER OF CHARACTERS ENTER MODE / VERIFY MODE		TOTAL ELAPSED TIME (HH:MM)	RECORDS PER HOUR (ENTER + VERIFY)	CHARACTERS PER HOUR (ENTER + VERIFY)
1 SUPERVISOR 1	156	0	4443	0	8:49	17	503
2 SUPERVISOR 2	0	0	0	0	0:00		
3 SUPERVISOR 3	0	0	0	0	0:00		
4 SUPERVISOR 4	0	0	0	0	0:00		
5 SUPERVISOR 5	0	0	0	0	0:00		
10 OPERATOR 10	0	0	0	0	0:00		
11 OPERATOR 11	0	0	0	0	0:00		
12 OPERATOR 12	0	0	0	0	0:00		
13 OPERATOR 13	0	0	0	0	0:00		
14 OPERATOR 14	0	0	0	0	0:00		
15 OPERATOR 15	0	0	0	0	0:00		
TOTALS	156	0	4443	0	8:49	17	503

Figure 6. Operator Report

DUMP FILE AND TANK FILE CONVERSION PROGRAMS

The structure of the Formats File may change between major releases of ODESY, as additional facilities become available. If this happens, the Dump File Conversion program must be used to convert existing formats so they are compatible with the new software level. Complete instructions for performing the conversion procedure, if required, are included in the documentation for each release. A Tank File Conversion program may also be used during the conversion procedure.

If the hardware or software fails during the conversion of formats, redump the format dump file from the old version and re-execute the program. To recover the Tank File Conversion program, reload backup copies of the original data files and re-execute the program.

An alternative use for the Dump File Conversion program is to produce a listing of all the groups and formats in a dump file. In this case, the formats are not changed, because they are already marked with the current release level. CONVERT can be executed while any other ODESY program is operating.

To use it, execute the Dump File Conversion program from the ODT. Enter and transmit:

```
EX ODESY/CONVERT
```

A message will be displayed on the ODT screen asking the name of the dump file to be converted or listed. Enter the dump file name via an accept message at the ODT. Use the mix number of the Dump File Conversion program. Enter and transmit:

```
<mix number>AX <format dump file name>
```

The program reads the file and produces a Dump File Conversion report showing all groups and formats in the dump file. As noted above, the formats will not be changed if they are already on the current release level. CONVERT then requests another dump file name.

To terminate the program, enter and transmit:

<mix number> AX

Figure 7 shows a sample Dump File Conversion report.

```
B1000 ON-LINE DATA ENTRY SYSTEM  D U M P F I L E   C O N V E R S I O N   R E P O R T  TIME 13:38  DATE 01/28/81  PAGE   1

  DUMPFILE:  ODY12/DUMPFILE

                    GROUP  FORMAT      DATE
                    MINT               11/17/80
                           MAND        11/17/80
```

Figure 7. Dump File Conversion Report

TANK FILE ANALYSIS PROGRAM

The Tank File Analysis program (DUMP) performs two functions that deal with a tank file in which corruption has been detected. The first function is the creation of a diagnostic report that provides information about the tank file. The second function enables you to manually reset the in-use flag for a batch. It can also be used to remove a corrupt batch that the squash procedure does not remove. These two functions require separate executions of the program.

THE REPORT FUNCTION

This function does not update the tank file but merely reads from it and creates the Tank File Analysis report. You may be alerted to the need for this report by the Edit program message:

```
74 PLEASE EXECUTE ODESY/DUMP
```

This occurs if EDIT encounters a fatal error during data entry. DUMP produces a diagnostic report which helps to identify the source of trouble. Send the dump report, the error report produced by EDIT at the time of the error, all three ODESYDATA files in machine-readable form, and your source and object-code files to your Burroughs System Support representative.

To create the Tank File Analysis (DUMP) report, execute the Tank File Analysis program from the ODT. Enter and transmit:

```
EX ODESY/DUMP
```

The program produces a Tank File Analysis report: a status report of the formats, system, and tank files, and a report of the tank file. This detailed report includes the status of each batch in the tank along with a listing of each record in the Tank File showing its status flags, addresses, group names, format names, pointers, and a postion of its data.

Distinguish DUMP, the Tank File Analysis program, from a Format Dump file, which is a serial file containing format data. A Format Dump file may not be used as a Formats file. It is accessed through the Format Maintenance program by the DUMP and LOAD commands.

THE BATCH FIXER FUNCTION

The second function available through the Tank File Analysis program enables you to manually reset a batch's in-use flag. You can also manually remove a corrupt batch from the tank file using this function. Deleting a batch in this manner does not free up the space occupied by the batch's records as would the Extraction program's delete or a tank file squash. Thus, it is best to use DUMP to delete a batch only as a last resort and follow the procedure with a tank file squash.

The batch fixer part of the Tank File Analysis program is activated by executing the program from the ODT with external switch 1 set to 1. Enter and transmit:

```
EX ODESY/DUMP . SW 1 1
```

This will cause DUMP to display the following message:

```
ENTER BATCH NUMBER, OR BLANK TO STOP RUNNING.
```

It will then accept the batch number of the batch to be accessed. Enter and transmit:

```
<mix number> AX  <batch number>
```

ODESY/DUMP will display the status of the specified batch. Check carefully to verify that the batch specified was the correct one. DUMP then displays the following instructions:

```
ENTER:  I       - TO TURN OFF THE IN-USE FLAG.
        R       - TO REMOVE BATCH FROM THE
                  TANK FILE.
      BLANK     - TO LEAVE STATUS UNCHANGED.
  NOTE:  IF 'R' IS ENTERED, THIS BATCH
         WILL NO LONGER EXIST TO ODESY.
```

At this point, enter the command desired, again using the ACCEPT keyboard command. If the batch number specified originally was incorrect, enter:

<mix number> AX

DUMP will re-display the original message. A new batch number may then be entered. If, however, the batch was correct, enter either I to reset the Batch's in-use flag, or R to permanently remove the batch from the tank file.

If R is specified, a final message is displayed to verify that the batch removal is desired. Enter YES to delete the batch or NO to leave the batch in the tank file.

When all updates have been made, enter a blank instead of a batch number to cause the program to go to end-of-job.

Figure 8 shows a sample Tank File Analysis report.

B1000 ON-LINE DATA ENTRY SYSTEM T A N K F I L E A N A L Y S I S R E P O R T TIME 10:49 DATE 01/23/81 PAGE 1

C O N T R O L R E C O R D D E T A I L S

FILE NAME	LAST RUN DATE	LAST RUN TIME	LAST REPORT DATE	REPORT NUMBER	FILE IN USE	PAGE SIZE	DATE TYPE	TANK FULL	EXTRACTION MATCH RECORD
FORMAT FILE	01/21/81	14:21			NO				
SYSTEM FILE	01/23/81	10:32	00/00/00	0	NO	60	MDY		
TANK FILE	01/23/81	10:32	01/20/81	15	NO			NO	

RECOVERY NEEDED	RECOVERY TYPE	CURRENT TIME	FIRST-FREE RECORD	FIRST-FREE BATCH	EOF POINTER	RECOVERY DETAILS STP	UPR RECORD	RECORD2	RECORD3
NO		09:56	235	29	256				

Figure 8. Tank File Analysis Report

SECTION 9

USER PROGRAM INTERFACE

The User Program Interface, sometimes referred to as Inquiry, is one of the most powerful features in ODESY. It allows the user to access the data during data entry in order to perform customized edit checks, virtually making any type of data editing possible. It may also reduce data entry requirements by allowing information supplied from the data base to appear in a screen format. If simple data base inquiry is desired and the transactions need not be saved, ODESY provides the option not to store each screen as a tank file record, for better disk utilization.

USES FOR THE USER PROGRAM INTERFACE

The User Program Interface is used in many different ways to fit a wide range of requirements. The following list shows the most common uses:

1. Simple inquiry into a file or data base.

2. Data file updating. When used this way, the data verification step is bypassed. However, it can be useful for some applications. You can still use the journal facility to provide an audit trail of these transactions.

3. Existence checking against a data file (for example, to verify that an account number exists).

4. Supplying information to the operator from a data file to reduce data entry.

5. Non-standard data checking, such as inter-field checking.

6. Other checking, such as when an unusually strict security system is needed. For example, the operator, format, and station can be checked against each other before the format is displayed.

The User Program Interface requires a special program, written by the user, to handle the customized checking and and data file accessing functions.

USER PROGRAM INTERFACE OPERATION

The user program interface works by allowing the user program to communicate with EDIT either before a format is displayed (for supplying information into the fields), after the format is diplayed (for additional checking of the data), or both. Errors found by the user program are handled as if they were found by Edit.

EDIT sends the data to the user program along with an accompanying header containing all the information pertaining to the data, such as the format name. One of the fields in the header contains the number of errors found by the user program. Most of the other fields in the header are not changed by the user program, but are for reference only.

The data string that follows the header corresponds with the layout of the format. The user program flags fields that are in error by moving question marks into them. A message type, along with the format name, tells the user program if it should supply information into the string or check the data. When this is accomplished, the user program returns the data with its header to EDIT. The remote file construct is used for this inter-program communication.

USING THE USER PROGRAM INTERFACE

If you do not require the User Program Interface, ignore all references to it. It will not affect the operation of the remainder of ODESY. Leave the system parameter Maximum Number of User Programs set to zero.

If, however, you can use this feature for a certain application, do the following:

1. Set the system parameter Maximum Number of User Programs to one or more.

2. You may want to select a relatively simple task, such as existence checking on an account number, for your first example.

3. Study Worked Example 1 at the end of this section, and model your user program after it.

4. Create the format for data entry. For this example, specify the format type in the format header as "C", a check-data type format. Indicate the user program name below it.

EDIT will handle this format in the following way when the operator requests it.

1. EDIT zip-executes the user program, then displays the format to the requesting operator.

2. The operator enters data, then transmits.

3. EDIT does all the standard checks on the data.

4. If no errors are found, EDIT sends the data, along with a message type of check ("03"), to the user program. (Otherwise, EDIT returns the data to the operator for correction.)

5. The user program should check the data, set the NUMBER OF ERRORS field in the header to the number of error fields found (if any), fill the fields in error with question marks, change the message type to data ("11"), and send it back to EDIT.

6. If the NUMBER OF ERRORS field is zero, EDIT stores the data in the tank file. Otherwise, it redisplays the screen, marking the fields in error.

NOTE

Unlike errors found by EDIT, error fields flagged by a user program are always displayed with question marks in the field, even when they are also blinked.

OVERVIEW OF INQUIRY FORMAT FUNCTIONS

This section begins with a general overview of inquiry formats. More detail on each of the formats will be provided later in the section.

There are four types of "inquiry" formats, or formats that require a user-written program. They are as follows:

1. Check-data.
2. Supply-data.
3. Key-data.
4. Supply-and-check.

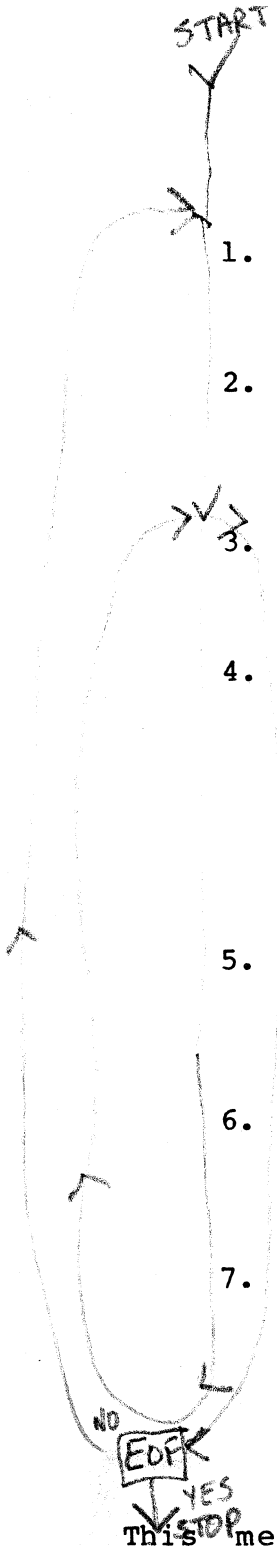
The check-data format allows the data to be sent to the user program after the screen is displayed and the data is entered by the operator. This format is designed for further checking of the data.

The supply-data format lets the user program supply information into fields before the format is displayed. For example, if an account number was entered previously, this format allows the user program to display the name and address of that account on the screen so the operator does not need to enter it. It can also be used for immediate visual verification of the account name. If a format needs data to be supplied to it before it is displayed, it should be specified as a supply-data type format.

The key-data format is used to send data that does not require immediate checking to a user program. In the above example, the "account number" could have been sent to the user program through a key-data format because the number is later checked when the following supply-data format is used. At that time, the user program will read a data file in order to supply the account information to the operator. If the "account number" key is found to be invalid, the user program can cause EDIT to redisplay the key-data format again, indicating an error, before the supply-data format appears on the screen.

The supply-and-check format is designed for tank file efficiency. It is important to remember that one record is stored in the tank file for each error-free data screen transmitted by an operator. (The only exception to this is when a batch is opened in Inquiry mode, in which case no records at all are stored.) It is important to use these formats efficiently so that records are not stored in the tank file unnecessarily. The supply-and-check format combines the functions of a supply-data format with that of a check-data format, so that both operations can be performed, while only one record is stored in the tank file.

A supply-and-check format, using the above inquiry example with account number as the key, would work like this:



1. The operator would request a key-data format to start the sequence. This would allow the operator to enter the first account number.

2. A supply-and-check format would be set up to follow automatically. The user program would supply the information about that account during the supply phase.

3. EDIT would then display the supply-and-check format to the operator, with information about the account appearing on the screen.

4. The operator would complete the entry of data by filling in the additional data required, if any, and then enter the next account number, probably at the very bottom of the screen. (To include the account number in the next record's data at Extract time, specify a "write" in the WRITE-AFTER option for the field just before "account number.")

5. After receiving the data, EDIT would do all of its standard checks on the data. Then if no errors were found, it would send the data to the user program.

6. During the check phase of the supply-and-check format, the user program would check the data further, if desired, and store the next account number.

7. The following format would be the same supply-and-check format. The cycle would repeat, starting with the supply phase in which the information pertaining to the previously-entered account number is supplied into the format and displayed to the operator.

This method allows only one tank file record to be stored for each transaction rather than one record for the account number and a second one for the information about that account.

The inquiry format types are specified to the Format Maintenance program on the format header screen as follows:

TYPE OF FORMAT	SPECIFICATION
Key-data	K
Supply-data	S
Check-data	C
Supply-and-check	U

The table below summarizes the sequence of events which occurs for each type of format. Normal, batch total, and display type formats are included for comparison.

	Blank or B	D	K	S	C	U
Select format	X	X	X	X	X	X
EDIT to user program				X		X
User program to EDIT				X		X
EDIT displays format	X	X	X	X	X	X
Operator enters data	X		X	X	X	X
EDIT checks data	X		X	X	X	X
EDIT to user program			X		X	X
User Program to EDIT					X	X
EDIT stores data (*)	X		X	X	X	X
Select next format	X	X	X	X	X	X

* If the batch is opened in Inquiry mode the data will not be stored.

You can select a format in one of four ways:

1. Select it initially by using a form of the ENTER command.
2. Select it automatically when a format is specified with a NEXT.FORMAT.

3. Override the "NEXT.FORMAT" field in the control line of the screen.

4. Use the user program to change the format name in the header. This may be done only during the supply phase, before the operator has seen the format. If the new format is a supply-type format, data sent from the user program will be displayed in it.

USER PROGRAM DESIGN

When writing a user program there are several design considerations to be aware of.

A user program can be written in any language that supports the remote file construct, which is used for interprogram communication. The examples at the back of this section show user programs written in COBOL, RPG, and UPL. DMSII data bases or other data files may be accessed by the user program. If an update on a data file is being done by the user program, however, careful consideration should be given to recovery. ODESY's recovery mechanisms are designed to recover only its ODESYDATA files, assuming those files are physically intact.

A user program communicates with EDIT through the remote file STATIONS. A remote file is a method of queueing messages. If you use COBOL 74, you must use the remote file construct rather than the communications section for inter-program communication. EDIT is an MCS-type program, and it uses the remote file to communicate with both terminals and user programs. Messages from a user program do not go directly to the terminals; they always go to EDIT. (EDIT sets the participating option when approving a file open.) If If the user program specifies a remote key, it will be ignored.

Do not execute a user program from the ODT. The program will be zip-executed by EDIT when a format that specifies it is selected.

EDIT brings down a user program by sending an end-of-file condition to it when it is no longer required. When this condition occurs, the communication between EDIT and the

program is broken and the user program should terminate gracefully.

The end-of-file condition is sent depending on the setting of the shutdown option in the system configuration. If the shutdown option is set to "C", which is the default, an end-of-file condition is sent to the user program when the CLOSE command is received from the last station using the program or when EDIT is terminated. If it is set to "T", the end-of-file condition is sent only when the data entry program is terminated.

EDIT?

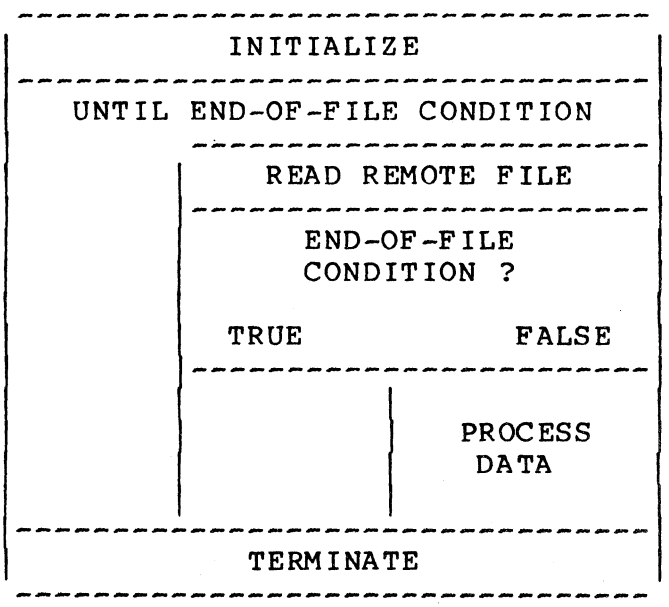
If you need more control over terminating the user program, you can set the shutdown option to "U". When the CLOSE command is received from the last station using the program, EDIT sends an advisory message to the user program. You can decide if the program should terminate depending upon factors such as beginning- and end-of-job overhead, memory utilization, frequency of utilization, or any other relevant considerations. You also can manually discontinue the user program by entering an ODESY DS command from a control station. This method is not usually recommended, because some other operator may be using the program when you bring it to EOJ.

NOTE

Only one user program should be active at a given station at one time. Thus, it is important to include procedures that handle at least all of the inquiry formats in a given group in one user program.

USER PROGRAM STRUCTURE

The structure of a user program normally is:



The main part of the program is a READ statement, followed by a set of procedure calls to process the message received and a return to the READ statement (see Inquiry Examples 1 and 2). The READ statement should have an AT END or ON.EOF branch because EDIT causes an end-of-file condition to terminate a user program.

The format of the messages in the remote file is described below.

MESSAGE HEADER

There is a standard form for all messages received and sent (READ and WRITE) by the user program. The message consists of a 30-byte header, which is followed by data. In COBOL nomenclature the message is:

```
 01 ODESY-RECORD-AREA.
1-2 03 ODESY-MSG-TYPE      PIC 99.
3-5 03 ODESY-STATION       PIC 999.
6-7 03 ODESY-OPERATOR      PIC 99.
```

```
8-10 03 ODESY-BATCH          PIC 999.
11-14 03 ODESY-RECORD        PIC 9999.
15-18 03 ODESY-GROUP         PIC XXXX.
19-22 03 ODESY-FORMAT        PIC XXXX.
23-26 03 ODESY-LENGTH        PIC 9999.
27-28 03 ODESY-ERRORS        PIC 99.
   29 03 ODESY-MODE          PIC X.
   30 03 ODESY-ACTION        PIC X.
      03 ODESY-DATA          PIC X(INPUT-DATA-SIZE).
```

The user program receives a message, and often sends a message in reply. Those parameters which are not to be changed in the reply are noted below.

Message Type

Messages with these message types are received by the user program from EDIT:

01 Key-data message.
02 Supply-data message.
03 Check-data message.
98 Last station using the program has closed the batch.

Messages with these message types are expected to be sent by the user program to EDIT:

11 Data message.
12 Invalid key message.
13 Invalid format message.

Station

This value is the logical station number (LSN) of the station that initiated the message. It must not be changed by the user program.

Operator

This number is that of the operator logged on at the station. It must not be changed by the user program.

Batch

This is the number of the batch that is open at the station. If Inquiry mode is being used, there is no batch open and this parameter is zero. It must not be changed by the user program.

Record

If a batch is open, this value is the relative record number of the record being entered, inserted, modified, or verified. It must not be changed by the user program.

Group

The group name is used in the OPEN command. It must not be changed by the user program.

Format

This option indicates the current format in use at the station. It can be changed only in reply to a supply-data message. Inquiry Example 2 shows this.

Length

This value describes the length of ODESY-DATA. It indicates the length of the data which follows the 30-byte header. Normally it does not need to be changed.

Errors

In incoming messages this value is always zero. In replying to a check-data message, the user program is expected to set this to the number of errors found, if any.

Mode

Mode indicates the current mode of the Edit Program. ODESY-MODE may assume these values, depending on how the batch was opened:

1 = Enter mode.
2 = Verify mode.
3 = Display mode.
4 = Inquiry mode.

The mode must not be changed by the user program.

Action

This indicates the current action in which EDIT is involved. ODESY-ACTION may assume these values, depending on the last command entered.

1 = Enter
2 = Verify
3 = Display
4 = Inquiry
5 = Modify
6 = Batch status
7 = Batch totals status

Data

In incoming key-data and check-data messages, this area contains all of the data for the record in the order that it was entered (that is, in the order that it was specified in the format). For suply-data messages, only default and duped-in values will be present in the data area sent to the user program.

For all message types, the data sent to the user program will be in character form. Data conversion, such as from character to packed, is done at extract time when the application file is created. The user program must return all of the data to EDIT in the same form, which is character form.

If the user program is replying to a check-data message, and errors were detected, the fields in error should be flagged with question marks (see Inquiry Examples 1 and 2).

The length of ODESY-DATA will never be greater than the value of the system parameter Input Data Size. However, for any particular message, the actual length of ODESY-DATA is given by ODESY-LENGTH.

INCOMING MESSAGES

The action taken by an user program after receiving a message usually depends on the group name, format name, and message type. The only rules enforced by ODESY refer to the message types of the messages sent in reply.

OUTGOING MESSAGES

The normal reply to a supply-data message or a check-data message is a data message (message type = 11). The data message either supplies data to EDIT or returns the result of checking the data.

However, if the user program sends an invalid key message (message type = 12) or an invalid format message (message type = 13), EDIT sends an error message to the operator. In the first case, it sends message 66 INVALID INQUIRY KEY and in the second case it sends message 67 INVALID INQUIRY FORMAT. If the user program sends an unrecognized message to EDIT, EDIT replies with message 68 INVALID INQUIRY REPLY.

INQUIRY FORMATS

There are four inquiry format types: key-data, supply-data, check-data, and supply-and-check data. When FORMAINT is being used to enter an inquiry format, a user program name is required. This is the name of the user program that will receive all messages for this format. A user program must be able to handle messages for all inquiry formats for which it is named. An inquiry format cannot be used if the relevant user program is not on disk.

KEY DATA

When a key-data format is used, a key-data message (message type = 01) is sent to the user program after successful checking of the data. The user program must not reply to this message.

The key-data format can be used for passing a key to the user program or for passing data to the user program for which no immediate checking is required. A key-data format would normally have a supply-data format as its NEXT.FORMAT, so that the information supplied for the latter relates to the key entered for the former. The user program may determine if the key passed by the key date format is valid when it receives the following supply data message.

If the key is valid, the user program may supply the data required (see explanation of supply data below). If the key is invalid, the user program can cause EDIT to redisplay the key-data format again before the supply-data format appears on the screen. This is done by changing the format name in the user program header back to the key-data format name.

If this method is to be used for inquiry, the user program should have a table with one entry for each station. The key-data supplied would be stored in this table for use with later supply-data or check-data messages. Format AP VEND in Inquiry Example 2 is an example of a key-data format.

SUPPLY DATA

When a supply-data format is used, a supply-data message (message type = 02) is sent to the user program before any data is entered. The user program is expected to reply with a data message (message type = 11), an invalid key message (message type = 12), or an invalid format message (message type = 13). If it is a data message, EDIT inspects the format name in the message, and if this format is a supply-data format or a supply-and-check format, the data at the end of the message is presented. The ability to change the format name allows the user program to alter the sequence of formats when there is no more data for this format or for this key. (See Inquiry Example 2, where the format is changed from AP INVC to AP VEND when there are no more invoices to supply.)

A supply-data format can be used for pure inquiry (no batch), or for interactive data entry where the data supplied will help the operator in entering a batch. Format AP INVC in Inquiry Example 2 is an example of a supply-data format.

NOTE

When a user program is initiated by a supply-data (or a supply-and-check) format, the request for the format must be sent to ODESY twice. The first request causes the user program to be zipped, but because communication with the user program is not possible until its remote file is opened, message 65 (INQUIRY PROGRAM IS NOT RUNNING) is displayed. After the second request, the format normally appears.

CHECK DATA

When a check-data format is used, a check-data message (message type = 03) is sent to the user program after successful checking by EDIT of the data. The user program is expected to reply with a data message (message type = 11) or an invalid key message (message type = 12) or an invalid format message (message type = 13).

If it is a data message, EDIT examines the ODESY-errors parameter in the message. If this parameter is greater than zero, the data in the message is presented to the operator for correction. If the ODESY-errors parameter is zero and there is a batch open (i.e., not Inquiry mode), the data in the message is stored in the batch as is. No further checking is done on the data by EDIT, because it was checked before being sent to the user program. Be sure to return the correct data after checking. In this way, the user program modifies the data before returning it to EDIT and this modified data is stored in the batch.

A check-data format can be used for on-line updating (no batch), for inter-field checks where the allowable values for one field depend on the value entered for another field, or for existence checking. Format INVC REC1 in Inquiry Example 1 is an example of a check-data format used for existence checking.

SUPPLY-AND-CHECK

When a supply-and-check format is used, a supply-data message (message type = 02) is sent to the user program before the operator is allowed to enter any data and a check-data message (message type = 03) is sent to the user program after successful checking by EDIT of the data.

A supply-and-check format combines the functions of a supply-data format and a check-data format. Format AP NAME in Inquiry Example 2 is an example of a supply-and-check format.

INQUIRY MODE

When inquiry formats are being used, batches are often unnecessary. Examples of this are pure inquiry, where there is no data entry, and on-line updating, where the user program uses the data entered directly to update a data base. In such cases, Inquiry mode can be used.

To get into Inquiry mode, enter the OPEN command followed by the keyword INQUIRY:

```
/OPEN INQUIRY <group>
```

You can now use the INQUIRY command to call up any format in the specified group:

```
//<format>
```

With the aid of the NEXT.FORMAT feature, you may be guided through an inquiry or update sequence. The AP formats in Inquiry Example 2 can be referenced in Inquiry mode.

To get out of Inquiry mode, enter a CLOSE command:

```
/CLOSE
```

Note that the Edit Program commands ACCEPT, DELETE, DISPLAY, ENTER, INSERT, MODIFY, and VERIFY can not be used in Inquiry mode.

EDIT COMMANDS WITH INQUIRY FORMATS

ACCEPT

This command is not affected by the format type. The record being entered or inserted is added to the batch, even though it may contain errors. However, no message is sent to the user program to indicate that the record has been accepted.

CLOSE

The CLOSE command's Effect on the user program depends, in part, upon the value of the system parameter SHUTDOWN. If the default SHUTDOWN option "C" is selected, the user program will receive an end-of-file (EOF) when the last batch using it is closed.

If the SHUTDOWN option has been set to "U" in the system configuration, an advisory message (message type = 98) is sent to the user program when the CLOSE command is received from the last station using it. The program can ignore the message, terminate immediately, or wait for a period of time before terminating. A complex wait statement may be used to wait for a period of time, and at the same time wait for messages from EDIT through the remote file.

If the SHUTDOWN option "T" is selected, the user program will not be notified in any way at CLOSE time, but will receive an EOF when EDIT terminates.

DELETE

This command is not affected by the format type. The record is deleted from the batch. However, no message is sent to the user program to indicate that the record has been deleted.

DISPLAY

This command is not affected by the format type. The data to be displayed is taken from the batch, even if it is a supply-data format or a supply-and-check format. No message is sent to the user program.

ENTER

For a supply-data format or a supply-and-check format, the user program is asked to supply data before entry. For a key-data format, a check-data format, or a supply-and-check format, the data is passed to the user program after entry.

INSERT

For a supply-data format or a supply-and-check format, the user program is asked to supply data before entry. For a key-data format, a check-data format, or a supply-and-check format, the data is passed to the user program after entry.

MODIFY

For a supply-data format or a supply-and-check format, the data presented for modification is taken from the batch. For a key-data format, a check-data format, or a supply-and-check format, the data is sent to the user program after entry. Thus key-data and check-data messages are sent, but supply-data messages are not sent.

VERIFY

For a supply-data format or a supply-and-check format, the data presented for verification is taken from the batch if the DISPLAY or NO VERIFY option is set or from the user program if the standard REKEY VERIFY option is set. If the format is a key-data format, a check-data format, or a supply-and-check format, the data is sent to the user program after entry. In this way key-data and check-data messages are sent and supply-data messages are sent if standard verify is specified for the format.

INQUIRY EXAMPLE 1

Inquiry Example 1 demonstrates how a simple existence check can be performed.

Format INVC REC1 is used for the entry of invoices. The stock number in each invoice is to be checked by the user program STOCKINQ against a file of valid stock numbers. For simplicity, the stock number is used directly as a key to access the stock file. The interface to ODESY would remain the same if a data base were being accessed.

To run this example under usercode security, you need a two-part name for the user program. See Usercode Security in Section 10.

FORMAT INVC REC1

The format is a check-data format and refers to the user program STOCKINQ. The stock number appears in bytes 6-10 of the input record.

DATA DIVISION (COBOL EXAMPLE)

The record area for the remote file is defined as 30 bytes of header information followed by 160 bytes of data. The 160 bytes of data is consistent with the Tank File Record Size option set to A. The record area is redefined so that it corresponds with the INVC REC1 format. The only important item in the redefinition is the stock number, which is to be checked.

PROCEDURE DIVISION (COBOL EXAMPLE)

During initialization, the stock file and the remote file are opened. The rest of the program is a loop to read the remote file and process the messages received.

If the format in the message is not INVC REC1, an invalid format message (message type = 13) is sent. Otherwise the stock number is checked against the stock file. If the number is valid, a data message (message type = 11) is sent,

with the data unchanged and ODESY-ERRORS = 0. If the stock number is invalid, a data message is still sent, but asterisks are used to indicate a stock number error in the data and ODESY-ERRORS = 1.

On end-of-file, for the remote file, the program terminates.

```
B1000 ON-LINE DATA ENTRY SYSTEM        F O R M A T   M A I N T E N A N C E   R E P O R T      TIME 15:22  DATE 01/27/81  PAGE   1
   GROUP NAME  INVC                                 FILE NAME            IC110                 DATE LAST CHANGED 01/21/81

   RECORD SIZE  80  BLOCKED  1   RECORDS/AREA  500   AREAS 40   ON DISK                                REPORT  DATE FORMAT  AS SYS
   GROUP SIGN CONVENTION     MSD


   FORMAT NAME INVCREC1          NEXT FORMAT INVCREC1   SECURITY  50      VERIFY   CHECK    DATE LAST CHANGED 01/21/81
                                 LINE ADVANCE 2         CURSOR ADVANCE  0
                                                                          USER PROGRAM NAME           STOCKINQ

                  ********************************************************************************
                  *  000 0000 REC1       [ ][REC1]                                               *
                  *                                                                              *
                  *     BATCH NO 2/3 [     ]                                                     *
                  *     STOCK NO     [               ]                                           *
                  *     ST.CODE      [A]         TR NO [     ]                                   *
                  *     QUANTITY     [         ]   D. F. [0]                                     *
                  *     VALUE        [           ]                                               *
                  *     DATE         [      ] REC.CODE    [2]                                    *
                  *     AUX CODE     [ ]                                                         *
                  *                                                                              *
                  ********************************************************************************
```

DISPLAY SPECIFICATIONS

SEQ	NAME	LENGTH	SPACE ADVANCE BEFORE	SPACE ADVANCE AFTER	LINE ADVANCE	HIGHLIGHT	DUP IN	DISP VER
10	BATCH NO 2/3	5	5	1	1		D	D
20	STOCK NO	15	5	5	1			D
30	ST.CODE	1	5	6	0			D
40	TR NO	5	8	1	1			D
50	QUANTITY	9	5	5	0			D
60	D. F.	1	4	1	1			D
70	VALUE	11	5	8	1			D
80	DATE	6	5	9	0		D	D
90	REC.CODE	1	1	4	1			D
100	AUX CODE	1	5	5	0			D

B1000 ON-LINE DATA ENTRY SYSTEM   F O R M A T   M A I N T E N A N C E   R E P O R T   TIME 15:22 DATE 01/27/81 PAGE   2

OTHER SPECIFICATIONS

SEQ	TYPE	CNS/VAL /RANGE	MAND	DATE	COV	CROSSFOOT	BATCH TOTALS	JUST IN	OUT POSN	WRITE AFTER	DUP OUT	JUST OUT	PACK OUT	BLANK FILL	SIGN	NAME COL	DATA COL	PRINT AFTER
10	UN		M						3			R		B			23	
20	UN		M					R	8			R		B			29	
30	AN	V	M						23							45	53	
			CONSTANTS A S U															
40	UN		M						24			R		B			56	
50	UN						1	L	29			R		B			62	
60	AN	V	M						38								72	
			CONSTANTS 0 1															
70	UN						2		39			R		B		74	80	
80	UN		M						30								92	
90	UN	V							1							1	10	
			CONSTANTS 2 3															
100	UN	V							2	C						12	21	C
			CONSTANTS 0 1 2															

```
**** TOTALS ****        FIELDS INPUT        10        FIELDS OUTPUT                          10
                        INPUT DATA SIZE     55        CHARACTERS OF CONSTANT DATA            11
                        OUTPUT DATA SIZE    55        CONSTANTS REQUIRED BY EXTRACT           0
                        FIELDS PRINTED      14        MINIMUM SCREEN SIZE OF FORMAT 12 BY 80
```

Figure 9-1. Format INVC REC1.

```
BURROUGHS B1800/B1700 COBOL COMPILER, MARK IX.0.0 (08/28/80 19:35)                    TUESDAY  , 01/27/81, 12:13:35
DDY20DEMOD/STOCKINQ.C

001000 $ CARD CHECK SINGLE CONTROL SUPPRESS
002000   IDENTIFICATION DIVISION.
003000 * ------------------------
004000   PROGRAM-ID. STOCKINQ
005000   AUTHOR. DATA ENTRY GROUP.
006000   DATE-WRITTEN. JULY 1980.
007000   DATE-COMPILED.
                       TIME 12:11 DATE 01/27/81.
008000 *
009000   ENVIRONMENT DIVISION.
010000 * --------------------
011000   CONFIGURATION SECTION.
012000   SOURCE-COMPUTER. B-1800.
013000   OBJECT-COMPUTER. B-1800.
014000   INPUT-OUTPUT SECTION.
015000   FILE-CONTROL.
016000       SELECT STOCK-FILE ASSIGN TO DISK.
017000              ACCESS MODE IS RANDOM,
018000              ACTUAL KEY IS STOCK-KEY.
019000       SELECT REMOTE-FILE ASSIGN TO REMOTE.
020000 /
021000   DATA DIVISION.
022000 * -------------
023000   FILE SECTION.
024000   FD   STOCK-FILE
025000        FILE CONTAINS 10 BY 500 RECORDS
026000        BLOCK CONTAINS 20 RECORDS
027000        RECORD CONTAINS 6 CHARACTERS
028000        VALUE OF ID IS "STOCKFILE".
029000   01   STOCK-REC.                                                    000.0000056
030000        03 STOCK-NO              PIC 9(5).                           000.0000056
031000        03 STOCK-VALID           PIC 9.
032000 *                                                                   000.0000066
033000   FD   REMOTE-FILE
034000        VALUE OF ID IS "STATIONS".
035000   01   ODESY-RECORD-AREA.                                           000.0000068
036000        03 ODESY-MSG-TYPE        PIC 99.                  (0001)     000.0000068
037000        03 ODESY-STATION         PIC 999.                            000.0000072
038000        03 ODESY-OPERATOR        PIC 99.                             000.0000078
039000        03 ODESY-BATCH           PIC 999.                            000.0000082
040000        03 ODESY-RECORD          PIC 9999.                           000.0000088
041000        03 ODESY-GROUP           PIC XXXX.                           000.0000096
042000        03 ODESY-FORMAT          PIC XXXX.                           000.0000104
043000        03 ODESY-LENGTH          PIC 9999.                           000.0000112
044000        03 ODESY-ERRORS          PIC 99.                             000.0000120
045000        03 FILLER                PIC XX.                             000.0000124
046000        03 ODESY-DATA            PIC X(160).                         000.0000128
047000   01   INVCREC1-AREA.                                               000.0000068
048000        03 FILLER                PIC X(30).                          000.0000068
049000        03 INVC-BATCH            PIC 9(5).                           000.0000128
050000        03 INVC-STOCK-NO         PIC 9(5).                (0002)     000.0000138
051000        03 INVC-ST-NO-ERRORS
052000           REDEFINES INVC-STOCK-NO PIC X(5).                         000.0000138
053000        03 INVC-STOCK-CODE       PIC X.                              000.0000148
054000        03 INVC-TR-NO            PIC 9(5).                           000.0000150
055000        03 INVC-QUANTITY         PIC 9(9).                           000.0000160
056000        03 INVC-DF               PIC X.                              000.0000178
057000        03 INVC-VALUE            PIC 9(11).                          000.0000180
058000        03 INVC-DATA             PIC 9(6).                           000.0000202
059000        03 INVC-REC-CODE         PIC 9.                              000.0000214
060000        03 INVC-AUX-CODE         PIC 9.
061000 *                                                                   000.0000216
062000   WORKING-STORAGE SECTION.
063000   77   STOCK-KEY               PIC 9(8) COMP.                      000.0000448
064000   77   INPUT-STOCK-RECORD      PIC X(9).
```

Figure 9-2. COBOL Compile Listing (1 of 2).

```
065000 /                                                                    000,0000456
066000   PROCEDURE DIVISION.
067000 * -------------------                                               000,0000000
068000   INITIALIZE.                                                        000,0000000
069000       OPEN I-O STOCK-FILE.                                           000,0000000
070000       OPEN I-O REMOTE-FILE.                                          000,0000082
071000   MAIN-LOOP.                                                         000,0000164
072000       READ REMOTE-FILE AT END STOP RUN.                              000,0000164
073000       IF ODESY-GROUP = "INVC" AND ODESY-FORMAT = "REC1"              000,0000491
074000          GO TO INVCREC1-CHECK.                                       000,0000573
075000 *     ELSE INVALID FORMAT                                            000,0000671
076000       MOVE 13 TO ODESY-MSG-TYPE.                                     000,0000671
077000 *                                                                   000,0000692
078000   WRITE-REMOTE.                                                      000,0000692
079000       WRITE ODESY-RECORD-AREA.                                       000,0000692
080000       GO TO MAIN-LOOP.                                               000,0000912
081000 *                                                                   000,0000928
082000   INVCREC1-CHECK.                                                    000,0000928
083000       MOVE INVC-STOCK-NO TO STOCK-KEY.                               000,0000928
084000       DISPLAY "INVCREC1-AREA IS " INVCREC1-AREA.                     000,0000961
085000       READ STOCK-FILE INVALID KEY GO TO INVCREC1-ERROR.              000,0001305
086000       DISPLAY "INVC-STOCK-NO IS " INVC-STOCK-NO.                     000,0001586
087000       IF STOCK-VALID = 0 GO TO INVCREC1-ERROR.                       000,0001932
088000       MOVE 11 TO ODESY-MSG-TYPE.                                     000,0001982
089000       GO TO WRITE-REMOTE.                                            000,0002003
090000 *                                                                   000,0002019
091000   INVCREC1-ERROR.                                                    000,0002019
092000       MOVE "*****" TO INVC-ST-NO-ERRORS.                             000,0002019
093000       MOVE 1 TO ODESY-ERRORS.                                        000,0002093
094000       MOVE 11 TO ODESY-MSG-TYPE.                                     000,0002131
095000       DISPLAY "STOCK-REC    = ".                                     000,0002152
096000       DISPLAY STOCK-REC.                                             000,0002436
097000       GO TO WRITE-REMOTE.                                            000,0002566
098000   END-OF-JOB.                                                        000,0002582


  C O D E    D I C T I O N A R Y
     BYTE LENGTH   CODEFILE RELATIVE DISK ADR
     ...........   ..........................
000  00000323      000007
     00000323      TOTAL CODE

  D A T A    D I C T I O N A R Y
     BYTE LENGTH   CODEFILE RELATIVE DISK ADR
     ...........   ..........................
000  00000392      000003

  S - M A C H I N E    P A R A M E T E R S    ( S C R A T C H P A D )

  LENB - 7, SEGB - 0, DISPB - 10, COPXB - 4, COPB - 22, D.E.F - 341, BDISPB - 12
  BASE RELATIVE ADDRESSES -
    DATA-SEGMENT-0 - 240,COP-TABLE - 42

  P R O G R A M    P A R A M E T E R    B L O C K

  FIRST-EXECUTABLE-INSTRUCTION 0,0
  INTERPRETER-NAME             COBOL     /INTERP
  STATIC-CORE                  3136 BITS
  DYNAMIC-CORE                 0 BITS
  DATA DICTIONARY STARTS AT CODEFILE SEGMENT 2, 1 ENTRY
  CODE DICTIONARY STARTS AT CODEFILE SEGMENT 6, 1 ENTRY
  FILE PARAMETER BLOCKS START AT CODEFILE SEGMENT 9, 2 ENTRIES

***** COMPILATION COMPLETE
ELAPSED TIME - 02 MINUTES, 55 SECONDS, WHICH IS 175 SECONDS
PROCESSOR TIME - 09 SECONDS, WHICH IS 9 SECONDS
PROGRAM REQUIRES 11 DISK SEGMENTS OF 180 BYTES EACH.
MEMORY REQUIREMENTS
  0000323 BYTES - LARGEST CODE SEGMENT
  0000392 BYTES - BASE-TO-LIMIT AREA
  0000315 BYTES - DICTIONARIES AND RUN STRUCTURE
  0000714 BYTES - FILE BUFFERS & FILE INFO AREAS - INCLUDES 129 BYTES (+72 TO 540 IF DISK) FOR EACH FILE
  0001744 BYTES - ESTIMATED MEMORY REQUIRED TO RUN IF ALL FILES OPEN
99 SYMBOLIC RECORDS COMPILED AT 33.900 RECORDS PER ELAPSED MINUTE OR 660.000 RECORDS PER PROCESSOR MINUTE.
```

Figure 9-3. COBOL Compile Listing (2 of 2).

```
B U R R O U G H S  B1800/B1700 R P G   C O M P I L E R ,   LEVEL -  MARK IX.1

 P R O G R A M   N A M E     ODY2ODEMOO/STOCKINQ.R
DATE COMPILED--/01/27/81     TIME COMPILED--13 10     WITH COMPILER-- DATED  08/21/80 11 18


          01000F*  1         2         3         4         5         6         7         8
          01100F*890123456789012345678901234567890123456789012345678901234567890123456789
          01200F*
          01300F*     ************************************************************
          01400F*     *                                                          *
          01500F*     *          S T O C K   I N Q U I R Y   P R O G R A M       *
          01600F*     *                                                          *
          01700F*     ************************************************************
          01800F*
          01900F*                    U S E   O F   I N D I C A T O R S
          02000F*
          02100F*          01  DUMMY PRIMARY FILE RECORD
          02200F*          02  STOCK FILE RECORD
          02300F*          03  REMOTE FILE RECORD
          02400F*
          02500F*          10  ODESY REQUEST FOR FORMAT INVCREC1
          02600F*
          02700F*          20  INVALID STOCK NUMBER
          02800F*          21  POSITIVE NON-ZERO STOCK NUMBER
          02900F*          22  EXCEPTION INDICATOR FOR REMOTE FILE WRITE
          03000F*
          03100F*          99  END OF FILE OR EXCEPTION ON REMOTE FILE
          03200F*
          03300F*
          03400F*     ************************************************************
          03500F*     *                                                          *
          03600F*     *        F I L E   S P E C I F I C A T I O N S             *
          03700F*     *                                                          *
          03800F*     ************************************************************
          03900F*
          04000F*  1         2         3         4         5         6         7         8
          04100F*890123456789012345678901234567890123456789012345678901234567890123456789
        1 04200FDUMMY    IPE F1600   80            DISK
          04300F* FAMILYSTOCKFILE
        2 04400FSTOCK    IC  F 120   6             DISK
          04500F* FAMILYSTATIONS
        3 04600FREMOTE   CD  F 190 190             REMOTE
          04700F*
          04800F*     ************************************************************
          04900F*     *                                                          *
          05000F*     *      T E L E C O M M U N I C A T I O N S   C A R D       *
          05100F*     *                                                          *
          05200F*     ************************************************************

        4 05300IREMOTE     10 10REMSTAREMLEN
```

Figure 9-4. RPG Compile Listing (1 of 3).

```
     054001*
     055001*          *************************************************************
     056001*          *                                                           *
     057001*          *           I N P U T   S P E C I F I C A T I O N S         *
     058001*          *                                                           *
     059001*          *************************************************************
     060001*
   5 061001DUMMY    NS  01
   6 062001                                                      1   1 FILLER
   7 063001STOCK    NS  02
   8 064001                                                      1   50STCKNO
   9 065001                                                      6   60STKOK          20
  10 066001REMOTE   NS  10   15 CI   16 CN   17 CV
  11 067001        AND       18 CC   19 CR   20 CE
  12 068001        AND       21 CC   22 C1
  13 069001                                                      1   200DYTYP
  14 070001                                                      3   500DYSTA
  15 071001                                                      6   700DYOPR
  16 072001                                                      8  1000DYBCH
  17 073001                                                     11  1400DYREC
  18 074001                                                     15  18 ODYGRP
  19 075001                                                     19  22 ODYFMT
  20 076001                                                     23  2600DYLEN
  21 077001                                                     27  2800DYERR
  22 078001                                                     31  350INVBCH
  23 079001                                                     36  400STKKEY        212020
  24 080001                                                     41  41 INVSCO
  25 081001                                                     42  460INVSTR
  26 082001                                                     47  550INVSQN
  27 083001                                                     56  56 INVDF
  28 084001                                                     57  672INVVAL
  29 085001                                                     68  730INVDAT
  30 086001                                                     74  740INVRCD
  31 087001                                                     75  750INVACD
  32 088001                                                      1 190 ODYMSG
  33 089001             NS  03
  34 090001                                                      1 190 ODYMSG


     09100C*
     09200C*          *************************************************************
     09300C*          *                                                           *
     09400C*          *   C A L C U L A T I O N   S P E C I F I C A T I O N S     *
     09500C*          *                                                           *
     09600C*          *************************************************************
     09700C*  1         2         3         4         5         6         7         8
     09800C*8901234567890123456789012345678901234567890123456789012345678901234567890
  35 09900C                 LOOP      TAG
  36 10000C                           SETOF                         202122
  37 10100C                           RECV REMOTE                   99   99
  38 10200C     99                    GOTO ENDIT
  39 10300C    N10                    MOVE 13         ODYTYP
  40 10400C     10  21      STKKEY    CHAINSTOCK                    20
  41 10500C     10  20                MOVE 1          ODYERR
  42 10600C     10                    MOVE 11         ODYTYP
  43 10700C                           SETON                         22
  44 10800C                           SEND REMOTE                        99
  45 10900C    N99                    GOTO LOOP
  46 11000C                 ENDIT     TAG
  47 11100C                           SETON                         LR

     112000*
     113000*          *************************************************************
     114000*          *                                                           *
     115000*          *         O U T P U T   S P E C I F I C A T I O N S         *
     116000*          *                                                           *
     117000*          *************************************************************
     118000*
  48 119000REMOTE  E         22
  49 119500                            ODYMSG   190
  50 120000                            ODYTYPX    2
  51 121000                            ODYERRX   28
  52 122000                  20                  40 '*****'

DIAGNOSTICS COMPLETE - NO SYNTAX ERRORS
```

Figure 9-5. RPG Compile Listing (2 of 3).

```
RPG COMPILER LEVEL  MARK IX.1. DATED 08/21/80

              NON-OVERLAYABLE MEMORY -    1123 BYTES
                  TOTAL FILE SPACE -  2721 BYTES -ASSUMING ALL FILES OPEN-
                  MINIMUM CODE SPACE -   431 BYTES
                                      ------
   PROGRAM DEPENDENT REQUIRED MEMORY -  4275 BYTES
            ESTIMATE OF MCP OVERHEAD -   200 BYTES
                                      ------
*** ESTIMATED MEMORY REQUIRED TO RUN -    4475 BYTES ***

OBJECT PROGRAM REQUIRES 35 DISK SEGMENTS OF 180 BYTES EACH
ELAPSED TIME 517 SECONDS
52 SOURCE CARDS COMPILED AT 6 CARDS PER MINUTE
```

Figure 9-6. RPG Compile Listing (3 of 3).

```
BURROUGHS B1800/B1700 UPL COMPILER, MARK IX.1 (10/03/80 22:46)     TUESDAY,    1/27/81, 12:17:00 PM

                          *****************************************
                          *                                       *
                          *             JOB NUMBER:               *
                          *               000776                  *
                          *                                       *
                          *                                       *
                          *            OBJECT FILE:               *
                          *       ODY20DEM00/STOCKINQ.U           *
                          *                                       *
                          *                                       *
                          *             CARDS FILE:               *
                          *        ODY20DEMS/STOCKINQ.U           *
                          *                                       *
                          *****************************************

LL  NL  SEQUENCE :SOURCE IMAGE                                                              :  PROCEDURE   SEGMENT    PATCH
 0   0  00002000 :%*********************************************************************:
 0   0  00003000 :%*                                                                    *:
 0   0  00004000 :%*     TITLE: B1000 ODESY                                             *:
 0   0  00005000 :%*            SAMPLE USER PROGRAM - STOCKINQ                          *:
 0   0  00006000 :%*                                                                    *:
 0   0  00007000 :%*********************************************************************:
 0   0  00008000 :%                                                                      :
 0   0  00009000 :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:
 0   0  00010000 :%                                                                     %:
 0   0  00011000 :%     PURPOSE      :  TO DEMONSTRATE HOW A SIMPLE EXISTENCE CHECK MAY BE %:
 0   0  00012000 :%                     PERFORMED BY A USER PROGRAM.                    %:
 0   0  00013000 :%                                                                     %:
 0   0  00014000 :%     INPUT        :  STOCK NUMBER FROM ODESY/EDIT, STOCKFILE CONTAINING %:
 0   0  00015000 :%                     ALL VALID STOCK NUMBERS.                        %:
 0   0  00016000 :%                                                                     %:
 0   0  00017000 :%     OUTPUT       :  STOCK NUMBER EITHER EXISTS OR NOT IN THE STOCKFILE %:
 0   0  00018000 :%                                                                     %:
 0   0  00019000 :%     PROCESS      :  ODESY/EDIT WILL ZIP EXECUTE THIS USER PROGRAM.  %:
 0   0  00020000 :%                     EACH FORMAT WILL BE CHECKED AS IT IS RECEIVED FROM %:
 0   0  00021000 :%                     EDIT. A REPLY WILL BE MADE BACK TO EDIT. THE    %:
 0   0  00022000 :%                     PROGRAM WILL TERMINATE WHEN EDIT SENDS AN EOF   %:
 0   0  00023000 :%                     MESSAGE FOR THE STATIONS FILE.                  %:
 0   0  00024000 :%                                                                     %:
 0   0  00025000 :%     ASSUMPTIONS :   FORMAT IS AS DESCRIBED IN THE INSTALLATION MANUAL %:
 0   0  00026000 :%                     EXAMPLE. STOCKFILE EXISTS WITH VALID STOCK NUMBERS.%:
 0   0  00027000 :%                                                                     %:
 0   0  00028000 :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:
 0   0  00029000 :%                                                                      :
 0   0  00030000 :%                                                                      :
 0   0  00031000 :%                                                                      :
 0   0  00032000 :%                                                                      :
 0   0  00033000 :DEFINE                                                                 :
 0   0  00034000 :    CH AS #CHARACTER#;                                                 :
 0   0  00035000 :%                                                                      :
 0   0  00036000 :FILE STOCKFILE (DEVICE = DISK RANDOM, RECORDS = 6/20, LOCK,            :
 0   0  00037000 :                     AREAS = 10/500, LABEL = "STOCKFILE");             :
 0   0  00038000 :FILE STATIONS  (DEVICE = REMOTE);                                      :
 0   0  00039000 :&PAGE                                                                  :
```

Figure 9-7. UPL Compile Listing (1 of 3).

```
0  0  :DECLARE                                                                  :
0  0  :          01 STOCK.REC,                                                  :
0  0  :             03 STOCK.NO           CH(5),                                :
0  0  :             03 STOCK.VALID        CH(1),                                :
0  0  :%                                                                        :
0  0  :          01 ODESY.RECORD.AREA,                                          :
0  0  :             03 ODESY.MSG.TYPE     CH(2),                                :
0  0  :             03 ODESY.STATION      CH(3),                                :
0  0  :             03 ODESY.OPERATOR     CH(2),                                :
0  0  :             03 ODESY.BATCH        CH(3),                                :
0  0  :             03 ODESY.RECORD       CH(4),                                :
0  0  :             03 ODESY.GROUP        CH(4),                                :
0  0  :             03 ODESY.FORMAT       CH(4),                                :
0  0  :             03 ODESY.LENGTH       CH(4),                                :
0  0  :             03 ODESY.ERRORS       CH(2),                                :
0  0  :             03 FILLER             CH(2),                                :
0  0  :             03 ODESY.DATA         CH(160),                              :
0  0  :%                                                                        :
0  0  :          01 INVCREC1.AREA REMAPS ODESY.RECORD.AREA,                     :
0  0  :             03 FILLER             CH(30),                               :
0  0  :             03 INVC.BATCH         CH(5),                                :
0  0  :             03 INVC.STOCK.NO      CH(5),                                :
0  0  :             03 INVC.STOCK.CODE    CH(1),                                :
0  0  :             03 INVC.TR.NO         CH(5),                                :
0  0  :             03 INVC.QUANTITY      CH(9),                                :
0  0  :             03 INVC.OF            CH(1),                                :
0  0  :             03 INVC.VALUE         CH(11),                               :
0  0  :             03 INVC.DATA          CH(6),                                :
0  0  :             03 INVC.REC.CODE      CH(1),                                :
0  0  :             03 INVC.AUX.CODE      CH(1),                                :
0  0  :%                                                                        :
0  0  :          STOCK.KEY                CH(5),                                :
0  0  :          STOCK.ADDR               FIXED;                                :
0  0  :%                                                                        :
0  0  :%                                                                        :
0  0  :PROCEDURE WRITE.REMOTE;                                                  :
1  0  :    WRITE STATIONS (ODESY.RECORD.AREA);                                  :   WRITE.REMO
1  0  :END WRITE.REMOTE;                                                        :   WRITE.REMO
0  0  :&PAGE                                                                    :
0  0  :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:
0  0  :%                                                                       %:
0  0  :%      WILL MOVE ASTERISKS TO THE STOCK NUMBER FIELD AND SET THE ERROR  %:
0  0  :%      FLAG TO TRUE. THESE ASTERISKS WILL APPEAR IN THE ODESY STATION   %:
0  0  :%      FORMAT SCREEN TO INDICATE THAT THE STOCK NUMBER IS NOT VALID.    %:
0  0  :%                                                                       %:
0  0  :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:
0  0  :PROCEDURE INVCREC1.ERROR;                                                :
1  0  :%                                                                        :   INVCREC1.E
1  0  :   INVC.STOCK.NO := "*****";                                             :   INVCREC1.E
1  0  :   ODESY.ERRORS := 1; % ERROR FLAG SET TO TRUE                           :   INVCREC1.E
1  0  :   ODESY.MSG.TYPE := 11;% DATA MESSAGE                                   :   INVCREC1.E
1  0  :END INVCREC1.ERROR;                                                      :   INVCREC1.E
0  0  :%                                                                        :
0  0  :PROCEDURE INVCREC1.CHECK;                                                :
1  0  :%                                                                        :   INVCREC1.C
1  0  :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:   INVCREC1.C
1  0  :%                                                                       %:   INVCREC1.C
1  0  :%    PURPOSE      :  TO CHECK FOR THE PRESENCE OF THE STOCK NUMBER IN   %:   INVCREC1.C
1  0  :%                    THE STOCK FILE.                                    %:   INVCREC1.C
1  0  :%                                                                       %:   INVCREC1.C
1  0  :%    INPUT        :  STOCK NUMBER FROM EDIT, STOCK FILE CONTAINING      %:   INVCREC1.C
1  0  :%                    ALL VALID STOCK NUMBERS, ODESY MESSAGE HEADER      %:   INVCREC1.C
1  0  :%                                                                       %:   INVCREC1.C
1  0  :%    OUTPUT       :  REPLY IN THE ODESY MESSAGE HEADER                  %:   INVCREC1.C
1  0  :%                                                                       %:   INVCREC1.C
1  0  :%    PROCESS      :  READ STOCK FILE USING THE STOCK NUMBER AS THE KEY. %:   INVCREC1.C
1  0  :%                    IF A STOCK NUMBER IS IN THE FILE, THEN IT WILL BE  %:   INVCREC1.C
1  0  :%                    TESTED TO SEE IF IT IS VALID. IF IT IS NOT IN THE  %:   INVCREC1.C
1  0  :%                    FILE, OR IF IT IS INVALID, CALL INVCREC1.ERROR,    %:   INVCREC1.C
1  0  :%                    OTHERWISE RETURN AN OK REPLY.                      %:   INVCREC1.C
1  0  :%                                                                       %:   INVCREC1.C
1  0  :%    ASSUMPTIONS :   STOCK FILE IS A RANDOM ACCESS FILE BUILT WITH THE  %:   INVCREC1.C
1  0  :%                    STOCK NUMBER AS THE KEY. IT CONTAINS THE STOCK     %:   INVCREC1.C
1  0  :%                    NUMBER AND A FLAG INDICATING IF IT IS VALID.       %:   INVCREC1.C
1  0  :%                                                                       %:   INVCREC1.C
1  0  :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:   INVCREC1.C
1  0  :%                                                                        :   INVCREC1.C
1  0  :   STOCK.ADDR := CONVERT(INVC.STOCK.NO,FIXED); % CONVERT CHAR TO NO.     :   INVCREC1.C
1  0  :   READ STOCKFILE [STOCK.ADDR] (STOCK.REC);                              :   INVCREC1.C
1  0  :        ON EOF % SHOULD NOT OCCUR ON RANDOM FILE                         :   INVCREC1.C
1  0  :               DO STOCKNO.NOT.IN.FILE;                                   :   INVCREC1.C
1  1  :                  INVCREC1.ERROR;                                        :   INVCREC1.C
1  1  :                  RETURN;                                                :   INVCREC1.C
1  1  :               END STOCKNO.NOT.IN.FILE;                                  :   INVCREC1.C
1  1  :        ON EXCEPTION % RECORD IS NOT IN THE STOCKFILE                    :   INVCREC1.C
1  0  :               DO STOCK.NO.NOT.IN.FILE;                                  :   INVCREC1.C
1  1  :                  INVCREC1.ERROR;                                        :   INVCREC1.C
1  1  :                  RETURN;                                                :   INVCREC1.C
1  1  :               END STOCK.NO.NOT.IN.FILE;                                 :   INVCREC1.C
1  1  :   IF STOCK.VALID = "0" THEN                                             :   INVCREC1.C
1  0  :      INVCREC1.ERROR; % BAD STOCK NUMBER                                 :   INVCREC1.C
1  1  :   ELSE ODESY.MSG.TYPE := "11";                                          :   INVCREC1.C
1  1  :END INVCREC1.CHECK;                                                      :   INVCREC1.C
0  0  :&PAGE                                                                    :
```

Figure 9-8. UPL Compile Listing (2 of 3).

```
0    0   00134000 :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:
0    0   00135000 :% EXECUTABLE CODE STARTS HERE                                            :
0    0   00136000 :    OPEN STOCKFILE INPUT,OUTPUT,LOCK;                                    :
0    0   00137000 :    OPEN STATIONS INPUT,OUTPUT;                                          :
0    0   00138000 :    DO MAIN.LOOP FOREVER;                                                :
0    1   00139000 :       READ STATIONS (INVCREC1.AREA);                                    :
0    1   00140000 :          ON EOF UNDO MAIN.LOOP; % TERMINATED BY EDIT                    :
0    2   00141000 :       IF ODESY.GROUP = "INVC" AND ODESY.FORMAT = "REC1" THEN            :
0    1   00142000 :         INVCREC1.CHECK;                                                 :
0    2   00143000 :       ELSE ODESY.MSG-TYPE := "13";                                      :
0    2   00144000 :       WRITE.REMOTE;                                                     :
0    1   00145000 :    END MAIN-LOOP;                                                       :
0    0   00146000 :FINI;                                                                    :

***** COMPILATION COMPLETE
```

```
COMPILE STATISTICS:

    NUMBER OF ERRORS DETECTED:      0
    NUMBER OF WARNINGS DETECTED:      0
    NUMBER OF CARDS SCANNED:      142
    NUMBER OF TOKENS SCANNED:       370
    LL ZERO NAME STACK ENTRIES:      27

PROGRAM STATISTICS

    CORE REQUIRED TO RUN:          1854 BYTES      (  14832 BITS)
    NUMBER OF SEGMENTS:            2
    SIZE OF LARGEST SEGMENT:        326 BYTES      (   2607 BITS)
    TOTAL SEGMENT SIZE:             326 BYTES      (   2607 BITS)
    DISK SIZE:                    11 SEGMENTS

RUN STATISTICS:

    NAME STACK SIZE:                 43 ENTRIES
    CONTROL STACK SIZE:             15 ENTRIES
    PROGRAM POINTER STACK SIZE:      25 ENTRIES
    EVALUATION STACK SIZE:          20 ENTRIES
    VALUE STACK SIZE:               10000 BITS
    PROGRAM STATIC CORE:              1834 BYTES      (  14672 BITS)
    PROGRAM DYNAMIC CORE:                0 BYTES      (      0 BITS)

COMPILE TIMES:

    ELAPSED TIME:    0:01:20.9
    PROCESSOR TIME:  08.4 SECONDS.
```

Figure 9-9. UPL Compile Listing (3 of 3).

INQUIRY EXAMPLE 2

Inquiry Example 2 demonstrates some of the more sophisticated inquiry facilities. This example examines selected records from a vendor master file and scans the vendor's outstanding invoices where necessary.

FORMATS AP VEND, AP NAME, AND AP INVC

Format AP VEND is a key-data format, format AP NAME is a supply-and-check format, and format AP INVC is a supply-data format. All formats refer to the user program APINQ.

DATA DIVISION (COBOL EXAMPLE)

The example shows record areas for the vendor master file, invoice file, and remote file. The record area for the remote file is redefined so that it corresponds with the data for the three formats. Also, there is a station table used to store keys for the vendor master file and the invoice file, which will accommodate up to 10 stations in this example.

PROCEDURE DIVISION (COBOL EXAMPLE)

During initialization, the vendor master file, invoice file, and remote file are opened, and the station table is set to zeros. The rest of the program is a loop to read the remote file and process the messages received.

If the message is a key-data message for format AP VEND, the vendor number is stored in the table as THIS-VEND and NEXT-VEND. No message is sent in reply.

If the message is a supply-data message for format AP NAME, NEXT-VEND is examined. If it contains a non-zero value, the vendor record is found and the relevant information is sent to EDIT in a data message. Format AP NAME requires 100 characters of data. If NEXT-VEND is zero, the format name is changed to AP INVC and the first invoice in the invoice file for THIS-VEND is found. The relevant information is sent to EDIT in a data message. Format AP INVC requires 88 characters of data.

If the message is a check-data message for format AP NAME, the value in the NEXT VENDOR field is stored in NEXT-VEND. If you have not entered a number, NEXT VENDOR will be blank. A data message is sent in reply, with ODESY-ERRORS = 0.

If the message is a supply-data message for format AP INVC, the next invoice for THIS-VEND is found. The relevant information is sent to EDIT in a data message. However, if there are no more invoices for THIS-VEND, the format name is changed to AP VEND and a data message, containing no data, is sent to EDIT.

OPERATION

You can select Inquiry mode or else open a batch in Enter mode to provide a journal. The journal shows all inquiries performed on the files.

After entering the /HELLO and /OPEN commands, enter //VEND.

The AP VEND format is displayed. Enter a vendor number.

The AP VEND data is sent to the user program in a key-data message. AP VEND is followed by AP NAME, so that a supply-data message is immediately sent to the user program. NEXT-VEND will be non-zero, so the vendor details are supplied in a data message.

The AP NAME format is displayed at the terminal with the vendor details.

If you wish to look at another vendor master record, enter a value for the field NEXT VENDOR. However, if you transmit without changing the data, the first invoice is supplied.

In either case, the AP NAME data is sent to the user program in a check-data message. The user program stores NEXT VENDOR as NEXT-VEND in its table and sends a data message in reply, with ODESY-ERRORS = 0.

The AP NAME format is followed by AP NAME, so another supply data message is immediately sent to the user program. If NEXT-VEND is zero, the format name is changed to AP INVC and the first invoice details are sent to EDIT in a data message.

The AP INVC format is displayed at the terminal with the vendor details.

If you transmit without changing the data, the next invoice is supplied. If you enter //NAME, the vendor details are recalled. You can examine another vendor master record by entering //VEND and then the vendor number.

If you choose to look at all of the invoices, the format will automatically change to AP VEND after the last invoice.

```
B1000 ON-LINE DATA ENTRY SYSTEM        F O R M A T  M A I N T E N A N C E  R E P O R T     TIME 15:22  DATE 01/27/81  PAGE   1
   GROUP NAME  AP                                      FILE NAME          NONE                 DATE LAST CHANGED 01/21/81
   RECORD SIZE 180  BLOCKED  1   RECORDS/AREA  500    AREAS 40    ON DISK                               REPORT  DATE FORMAT  AS SYS
   GROUP SIGN CONVENTION      MSD

   FORMAT NAME AP  VEND            NEXT FORMAT AP  NAME    SECURITY  80        VERIFY   KDATA    DATE LAST CHANGED 01/21/81
                                   LINE ADVANCE 3          CURSOR ADVANCE  0
                                                                               USER PROGRAM NAME            APINQ

                    **************************************************************************
                    *  000 0000 VEND       [ ][NAME]                                         *
                    *                                                                        *
                    *     VENDOR NUMBER  [       ]                                           *
                    *                                                                        *
                    *                                                                        *
                    *                                                                        *
                    *                                                                        *
                    *                                                                        *
                    **************************************************************************
DISPLAY SPECIFICATIONS
                                                   SPACE ADVANCE  LINE      HIGHLIGHT  DUP   DISP
SEQ        NAME                            LENGTH  BEFORE/AFTER  ADVANCE               IN    VER
 10  VENDOR NUMBER                            6       5     2       1


OTHER SPECIFICATIONS
SEQ  TYPE  CNS/VAL  MAND  DATE  CDV  CROSSFOOT   BATCH    JUST  OUT   WRITE  DUP   JUST  PACK  BLANK  SIGN  NAME   DATA  PRINT
           /RANGE                                TOTALS   IN    POSN  AFTER  OUT   OUT   OUT   FILL         COL    COL   AFTER
 10   UN             M                                    R
**** WARNING ****  THIS FIELD IS NEITHER OUTPUT NOR PRINTED

**** WARNING ****  THE BUFFER IS NOT WRITTEN AFTER LAST FIELD
**** TOTALS  ****          FIELDS INPUT        1          FIELDS OUTPUT                     0
                           INPUT DATA SIZE     6          CHARACTERS OF CONSTANT DATA       0
                           OUTPUT DATA SIZE    6          CONSTANTS REQUIRED BY EXTRACT     0
                           FIELDS PRINTED      0          MINIMUM SCREEN SIZE OF FORMAT 8 BY 32
```

Figure 9-10. Formats AP VEND, AP NAME, and AP INVC (1 of 4).

FORMAT NAME AP NAME NEXT FORMAT AP NAME SECURITY 80 VERIFY C-&-S DATE LAST CHANGED 01/21/81
LINE ADVANCE 3 CURSOR ADVANCE 0
USER PROGRAM NAME APINQ

```
000 0000 NAME      [ ][NAME]


    VENDOR NUM  [      ]
                [                    ]
                [                    ]
                [                    ]
    PAYBLE BAL  [          ]
    LST ACTIVE  [      ]
    NEXT VEND   [      ]
```

DISPLAY SPECIFICATIONS

SEQ	NAME	LENGTH	SPACE ADVANCE BEFORE/AFTER		LINE ADVANCE	HIGHLIGHT	DUP IN	DISP VER
10	VENDOR NUM	6	5	2	1			
20		24	16	1	1			
30		24	16	1	1			
40		24	16	1	1			
50	PAYBLE BAL	10	5	2	1			
60	LST ACTIVE	6	5	2	1			
70	NEXT VEND	6	5	3	0			

OTHER SPECIFICATIONS

SEQ	TYPE	CNS/VAL /RANGE	MAND	DATE	CDV	CROSSFOOT	BATCH TOTALS	JUST IN	OUT POSN	WRITE AFTER	DUP OUT	JUST OUT	PACK OUT	BLANK FILL	SIGN	NAME COL	DATA COL	PRINT AFTER
10	UN															1	12	
20	AN																20	
30	AN																50	
40	AN																80	C

OTHER SPECIFICATIONS

SEQ	TYPE	CNS/VAL /RANGE	MAND	DATE	CDV	CROSSFOOT	BATCH TOTALS	JUST IN	OUT POSN	WRITE AFTER	DUP OUT	JUST OUT	PACK OUT	BLANK FILL	SIGN	NAME COL	DATA COL	PRINT AFTER
50	UN															20	31	
60	UN															50	61	
70	UN							R										C

**** WARNING **** THIS FIELD IS NEITHER OUTPUT NOR PRINTED

**** WARNING **** THE BUFFER IS NOT WRITTEN AFTER LAST FIELD

**** TOTALS ****

FIELDS INPUT	7	FIELDS OUTPUT	0
INPUT DATA SIZE	100	CHARACTERS OF CONSTANT DATA	0
OUTPUT DATA SIZE	100	CONSTANTS REQUIRED BY EXTRACT	0
FIELDS PRINTED	9	MINIMUM SCREEN SIZE OF FORMAT	12 BY 80

Figure 9-11. Formats AP VEND, AP NAME, and AP INVC (2 of 4).

```
B1000 ON-LINE DATA ENTRY SYSTEM        F O R M A T   M A I N T E N A N C E   R E P O R T      TIME 15:22  DATE 01/27/81  PAGE   6

   FORMAT NAME AP  INVC           NEXT FORMAT AP  INVC    SECURITY  80        VERIFY   SUPP     DATE LAST CHANGED 01/21/81
                                  LINE ADVANCE 3          CURSOR ADVANCE  0
                                                                              USER PROGRAM NAME              APINQ

                      ****************************************************************************
                      *  000 0000 INVC        [ ][INVC]                                           *
                      *                                                                          *
                      *                                                                          *
                      *     VENDOR NUM [        ]                                                 *
                      *     TRAN NUM   [        ] INV NUM   [          ]                          *
                      *     DATE       [        ] DUE DATE  [        ]                            *
                      *     AMOUNT [            ]                                                 *
                      *     PAID   [            ] PAYABLE   [          ]                          *
                      *            [                          ]                                   *
                      *                                                                          *
                      ****************************************************************************
```

DISPLAY SPECIFICATIONS

SEQ	NAME	LENGTH	SPACE BEFORE	ADVANCE AFTER	LINE ADVANCE	HIGHLIGHT	DUP IN	DISP VER
10	VENDOR NUM	6	5	1	1			
20	TRAN NUM	6	5	3	0			
30	INV NUM	10	1	3	1			
40	DATE	6	5	7	0			
50	DUE DATE	6	1	2	1			
60	AMOUNT	10	5	1	1			
70	PAID	10	5	3	0			
80	PAYABLE	10	1	3	1			
90		24	11	1	0			

OTHER SPECIFICATIONS

SEQ	TYPE	CNS/VAL /RANGE	MAND	DATE	CDV	CROSSFOOT	BATCH TOTALS	JUST IN	JUT POSN	WRITE AFTER	DUP OUT	JUST OUT	PACK OUT	BLANK FILL	SIGN	NAME COL	DATA COL	PRINT AFTER
10	UN																	

**** WARNING **** THIS FIELD IS NEITHER OUTPUT NOR PRINTED

Figure 9-12. Formats AP VEND, AP NAME, and AP INVC (3 of 4).

B1000 ON-LINE DATA ENTRY SYSTEM F O R M A T M A I N T E N A N C E R E P O R T TIME 15:22 DATE 01/27/81 PAGE 5

OTHER SPECIFICATIONS

SEQ	TYPE	CNS/VAL /RANGE	MAND	DATE	CDV	CROSSFOOT	BATCH TOTALS	JUST IN	DUP POSN	WRITE AFTER	DUP OUT	JUST OUT	PACK OUT	BLANK FILL	SIGN	NAME COL	DATA COL	PRINT AFTER
20	UN																20	
30	AN																30	
40	UN															44	49	
50	UN															59	68	C
60	UN															35	42	
70	UN															56	61	
80	UN															75	83	
90	AN																97	C

**** WARNING **** THE BUFFER IS NOT WRITTEN AFTER LAST FIELD

**** TOTALS ****

FIELDS INPUT	9	FIELDS OUTPUT	0
INPUT DATA SIZE	80	CHARACTERS OF CONSTANT DATA	0
OUTPUT DATA SIZE	80	CONSTANTS REQUIRED BY EXTRACT	0
FIELDS PRINTED	13	MINIMUM SCREEN SIZE OF FORMAT	12 BY 80

Figure 9-13. Formats AP VEND, AP NAME, and AP INVC (4 of 4).

```
BURROUGHS B1800/B1700 COBOL COMPILER, MARK IX.0.0 (08/28/80 19:35)                    TUESDAY  , 01/27/81, 12:14:09
JDY20DEMO0/APINQ.C

001000 $ CARD CHECK SINGLE CONTROL SUPPRESS
002000   IDENTIFICATION DIVISION.
003000 * ------------------------
004000   PROGRAM-ID. APINQ.
005000   AUTHOR. DATA ENTRY GROUP.
006000   DATE-WRITTEN. JULY 1980.
007000   DATE-COMPILED.
                        TIME 12:10 DATE 01/27/81.
008000 *
009000   ENVIRONMENT DIVISION.
010000 * ---------------------
011000   CONFIGURATION SECTION.
012000   SOURCE-COMPUTER. B-1800.
013000   OBJECT-COMPUTER. B-1800.
014000   INPUT-OUTPUT SECTION.
015000   FILE-CONTROL.
016000       SELECT VENDOR-MASTER ASSIGN TO DISK,
017000              ACCESS MODE IS RANDOM,
018000              ACTUAL KEY IS VENDOR-KEY.
019000       SELECT INVOICE-FILE ASSIGN TO DISK,
020000              ACCESS MODE IS RANDOM,
021000              ACTUAL KEY IS INVOICE-KEY.
022000       SELECT REMOTE-FILE ASSIGN TO REMOTE.
023000 /
024000   DATA DIVISION.
025000 * --------------
026000   FILE SECTION.
027000   FD   VENDOR-MASTER
028000        FILE CONTAINS 2000 BY 500 RECORDS,
029000        BLOCK CONTAINS 20 RECORDS,
030000        RECORD CONTAINS 187 CHARACTERS,
031000        VALUE OF ID IS "APVNMR".
032000   01   VENDOR-REC-AREA.                                                  000,0000056
033000        03 VEND-VENDNO                PIC 9(6) COMP.                     000,0000056
034000        03 VEND-NAME                  PIC X(24).                         000,0000062
035000        03 VEND-ADDR                  PIC X(24).                         000,0000110
036000        03 VEND-ADDR2                 PIC X(24).                         000,0000158
037000        03 FILLER                     PIC X(16).                         000,0000206
038000        03 VEND-PAYBAL                PIC S9(9) COMP.          (0001)    000,0000238
039000        03 FILLER                     PIC X(35).                         000,0000248
040000        03 VEND-DATE-LA               PIC 9(6).                          000,0000318
041000        03 FILLER                     PIC X(43).                         000,0000330
042000   FD   INVOICE-FILE
043000        FILE CONTAINS 2000 BY 500 RECORDS,
044000        BLOCK CONTAINS 20 RECORDS,
045000        RECORD CONTAINS 104 CHARACTERS,
046000        VALUE OF ID IS "APVNIN".
047000   01   INVOICE-REC-AREA.                                                 000,0000416
048000        03 INVC-VENDNO                PIC 9(6) COMP.           (0002)    000,0000416
049000        03 INVC-TRANNO                PIC 9(6) COMP.                     000,0000422
050000        03 INVC-INVNO                 PIC X(10).                         000,0000428
051000        03 INVC-DATE                  PIC 9(6).                          000,0000448
052000        03 INVC-DUE-DATE              PIC 9(6).                          000,0000460
053000        03 INVC-AMOUNT                PIC S9(9) COMP.          (0003)    000,0000472
054000        03 INVC-PAID                  PIC S9(9) COMP.                    000,0000482
055000        03 INVC-PAYABLE               PIC S9(9) COMP.          (0004)    000,0000492
056000        03 FILLER                     PIC X(19).                         000,0000502
057000        03 INVC-DESC                  PIC X(24).                         000,0000540
058000        03 FILLER                     PIC X(4).
```

Figure 9-14. COBOL Compile Listing (1 of 5).

```
      /                                                                                    000,0000588
        FD   REMOTE-FILE
             VALUE OF ID IS "STATIONS".
        01   ODESY-RECORD-AREA.                                                    (0005) 000,0000596
             03 ODESY-MSG-TYPE                PIC 99.                              (0006) 000,0000596
             03 ODESY-STATION                 PIC 999.                             (0007) 000,0000600
             03 ODESY-OPERATOR                PIC 99.                                     000,0000606
             03 ODESY-BATCH                   PIC 999.                                    000,0000610
             03 ODESY-RECORD                  PIC 9999.                                   000,0000616
             03 ODESY-GROUP                   PIC XXXX.                                   000,0000624
             03 ODESY-FORMAT                  PIC XXXX.                            (0008) 000,0000632
             03 ODESY-LENGTH                  PIC 9999.                            (0009) 000,0000640
             03 ODESY-ERRORS                  PIC 99.                                     000,0000648
             03 FILLER                        PIC XX.                                     000,0000652
             03 ODESY-DATA                    PIC X(160).                          (0010) 000,0000656
        01   APVEND-AREA.                                                                 000,0000596
             03 FILLER                        PIC X(30).                                  000,0000596
             03 APVEND-VENDNO                 PIC 9(6).                            (0011) 000,0000656
        01   APNAME-AREA.                                                                 000,0000596
             03 FILLER                        PIC X(30).                                  000,0000596
             03 APNAME-VENDNO                 PIC 9(6).                            (0012) 000,0000656
             03 APNAME-NAME                   PIC X(24).                           (0013) 000,0000668
             03 APNAME-ADDR                   PIC X(24).                                  000,0000716
             03 APNAME-ADDR2                  PIC X(24).                                  000,0000764
             03 APNAME-SPAYBAL                PIC X.                                      000,0000812
             03 APNAME-PAYBAL                 PIC 9(9).                                   000,0000814
             03 APNAME-DATE                   PIC 9(6).                                   000,0000832
             03 APNAME-NEXTVEND               PIC 9(6).                                   000,0000844
        01   APINVC-AREA.                                                                 000,0000596
             03 FILLER                        PIC X(30).                                  000,0000596
             03 APINVC-VENDNO                 PIC 9(6).                                   000,0000656
             03 APINVC-TRANNO                 PIC 9(6).                                   000,0000668
             03 APINVC-INVNO                  PIC X(10).                                  000,0000680
             03 APINVC-DATE                   PIC 9(6).                                   000,0000700
             03 APINVC-DUE-DATE               PIC 9(6).                                   000,0000712
             03 APINVC-SAMOUNT                PIC X.                                      000,0000724
             03 APINVC-AMOUNT                 PIC 9(9).                                   000,0000726
             03 APINVC-SPAID                  PIC X.                                      000,0000744
             03 APINVC-PAID                   PIC 9(9).                                   000,0000746
             03 APINVC-SPAYABLE               PIC X.                                      000,0000764
             03 APINVC-PAYABLE                PIC 9(9).                                   000,0000766
             03 APINVC-DESC                   PIC X(24).
      *                                                                                   000,0000784
        WORKING-STORAGE SECTION.
        77   VENDOR-KEY                       PIC 9(8) COMP.                              000,0000976
        77   INVOICE-KEY                      PIC 9(8) COMP.                       (0014) 000,0000984
        77   I                                PIC 99 COMP.                         (0015) 000,0000992
        01   STATION-TABLE.                                                               000,0000994
             03 STATION-ENTRY  OCCURS 10 TIMES.                                           000,0000994
                05 THIS-VEND                  PIC 9(6) COMP.                       (0016) 000,0000994
                05 NEXT-VEND                  PIC 9(6) COMP.                       (0018) 000,0001000
                05 THIS-INV                   PIC 9(8) COMP.
```

Figure 9-15. COBOL Compile Listing (2 of 5).

```
111000 /                                                                        (0020) 000.0001006
112000   PROCEDURE DIVISION.
113000 * -------------------                                                           000.0000000
114000   INITIALIZE.                                                                  000.0000000
115000       OPEN INPUT VENDOR-MASTER.                                                000.0000000
116000       OPEN INPUT INVOICE-FILE.                                                 000.0000084
117000       OPEN I-O REMOTE-FILE.                                                    000.0000168
118000       MOVE 1 TO I.                                                             000.0000252
119000   INIT-STATION-TABLE-LOOP.                                                     000.0000270
120000       MOVE 0 TO THIS-VEND(I).                                                  000.0000270
121000       MOVE 0 TO NEXT-VEND(I).                                                  000.0000289
122000       MOVE 0 TO THIS-INV(I).                                                   000.0000308
123000       ADD 1 TO I.                                                              000.0000327
124000       IF I < 11 GO TO INIT-STATION-TABLE-LOOP.                                 000.0000341
125000   MAIN-LOOP.                                                                   000.0000360
126000       READ REMOTE-FILE AT END STOP RUN.                                        000.0000380
127000       IF ODESY-GROUP = "AP" GO TO AP-CHECK.                                    000.0000720
128000   INVALID-FORMAT.                                                              000.0000790
129000       DISPLAY "INVALID FORMAT".                                                000.0000790
130000       MOVE 13 TO ODESY-MSG-TYPE.                                               000.0001079
131000       WRITE ODESY-RECORD-AREA.                                                 000.0001101
132000       GO TO MAIN-LOOP.                                                         000.0001327
133000   INVALID-KEY.                                                                 000.0001344
134000       DISPLAY "INVALID KEY".                                                   000.0001344
135000       MOVE 12 TO ODESY-MSG-TYPE.                                               000.0001609
136000       WRITE ODESY-RECORD-AREA.                                                 000.0001631
137000       GO TO MAIN-LOOP.                                                         000.0001857
138000   WRITE-REMOTE.                                                                000.0001874
139000       MOVE 11 TO ODESY-MSG-TYPE.                                               000.0001874
140000       WRITE ODESY-RECORD-AREA.                                                 000.0001896
141000       GO TO MAIN-LOOP.                                                         000.0002122
142000   AP-CHECK.                                                                    000.0002139
143000       IF ODESY-FORMAT = "VEND" GO TO APVEND-KEYDATA.                           000.0002139
144000       IF THIS-VEND(ODESY-STATION) = 0 GO TO INVALID-KEY.                       000.0002202
145000       IF ODESY-FORMAT = "NAME" AND ODESY-MSG-TYPE = 2                          000.0002238
146000          GO TO APNAME-SUPPLY.                                                  000.0002301
147000       IF ODESY-FORMAT = "NAME" AND ODESY-MSG-TYPE = 3                          000.0002353
148000          GO TO APNAME-CHECK.                                                   000.0002416
149000       IF ODESY-FORMAT = "INVC"                                                 000.0002468
150000          GO TO APINVC-SUPPLY.                                                  000.0002468
151000       GO TO INVALID-FORMAT.                                                    000.0002531
```

Figure 9-16. COBOL Compile Listing (3 of 5).

```
152000 /                                                                        000.0002548
153000   APVEND-KEYDATA.                                                        000.0002548
154000       DISPLAY "APVEND KEYDATA".                                          000.0002548
155000       MOVE APVEND-VENDNO TO THIS-VEND(ODESY-STATION).                    000.0002837
156000       MOVE APVEND-VENDNO TO NEXT-VEND(ODESY-STATION).                    000.0002856
157000       MOVE 0 TO THIS-INV(ODESY-STATION).                                 000.0002875
158000       GO TO MAIN-LOOP.                                                   000.0002894
159000   APNAME-SUPPLY.                                                         000.0002911
160000       DISPLAY "APNAME SUPPLY".                                           000.0002911
161000       IF NEXT-VEND(ODESY-STATION) = 0                                    000.0003192
162000 *        THEN SUPPLY FIRST INVOICE FOR THIS VENDOR                       000.0003192
163000 *        GO TO APNAME-FIRST-INVOICE.                                     000.0003192
164000 *     ELSE SUPPLY NEXT VENDOR DETAILS                                    000.0003228
165000       MOVE NEXT-VEND(ODESY-STATION) TO THIS-VEND(ODESY-STATION).         000.0003228
166000       MOVE THIS-VEND(ODESY-STATION) TO VENDOR-KEY.                       000.0003252
167000       READ VENDOR-MASTER INVALID KEY GO TO MISSING-VENDOR.               000.0003294
168000       MOVE SPACES TO ODESY-DATA.                                         000.0003589
169000       MOVE VEND-VENDNO TO APNAME-VENDNO.                                 000.0003603
170000       MOVE VEND-NAME TO APNAME-NAME.                                     000.0003640
171000       MOVE VEND-ADDR TO APNAME-ADDR.                                     000.0003677
172000       MOVE VEND-ADDR2 TO APNAME-ADDR2                                    000.0003737
173000       MOVE VEND-PAYBAL TO APNAME-PAYBAL                                  000.0003737
174000       IF VEND-PAYBAL < 0 MOVE "-" TO APNAME-SPAYBAL.                     000.0003797
175000       MOVE VEND-DATE-LA TO APNAME-DATE.                                  000.0003910
176000       MOVE 100 TO ODESY-LENGTH.                                          000.0003970
177000       GO TO WRITE-REMOTE.                                                000.0003996
178000   MISSING-VENDOR.                                                        000.0004013
179000       DISPLAY "MISSING VENDOR".                                          000.0004013
180000       MOVE SPACES TO ODESY-DATA.                                         000.0004302
181000       MOVE THIS-VEND(ODESY-STATION) TO APNAME-VENDNO.                    000.0004316
182000       MOVE "MISSING VENDOR" TO APNAME-NAME.                              000.0004335
183000       MOVE 100 TO ODESY-LENGTH.                                          000.0004469
184000       GO TO WRITE-REMOTE.                                                000.0004495
185000   APNAME-FIRST-INVOICE.                                                  000.0004512
186000       MOVE THIS-VEND(ODESY-STATION) TO NEXT-VEND(ODESY-STATION).         000.0004512
187000 *     CHANGE FORMAT NAME                                                 000.0004536
188000       MOVE "INVC" TO ODESY-FORMAT.                                       000.0004536
189000       MOVE 0 TO THIS-INV(ODESY-STATION).                                 000.0004582
190000       GO TO APINVC-SUPPLY.                                               000.0004601
191000   APNAME-CHECK.                                                          000.0004618
192000       DISPLAY "APNAME CHECK".                                            000.0004618
193000       MOVE APNAME-NEXTVEND TO NEXT-VEND(ODESY-STATION).                  000.0004891
194000       GO TO WRITE-REMOTE.                                                000.0004933
195000 /                                                                        000.0004950
196000   APINVC-SUPPLY.                                                         000.0004950
197000       DISPLAY "APINVC SUPPLY".                                           000.0004950
198000       MOVE THIS-INV(ODESY-STATION) TO INVOICE-KEY.                       000.0005231
199000   READ-INVOICE.                                                          000.0005250
200000       ADD 1 TO INVOICE-KEY.                                              000.0005250
201000       READ INVOICE-FILE INVALID KEY GO TO LAST-INVOICE.                  000.0005264
202000       IF THIS-VEND(ODESY-STATION) NOT = INVC-VENDNO                      000.0005536
203000 *        THEN READ NEXT INVOICE                                          000.0005536
204000 *        GO TO READ-INVOICE.                                             000.0005536
205000 *     ELSE SUPPLY INVOICE DETAILS                                        000.0005572
206000       MOVE INVOICE-KEY TO THIS-INV(ODESY-STATION).                       000.0005572
207000       MOVE SPACES TO ODESY-DATA.                                         000.0005591
208000       MOVE INVC-VENDNO TO APINVC-VENDNO.                                 000.0005605
209000       MOVE INVC-TRANNO TO APINVC-TRANNO.                                 000.0005642
210000       MOVE INVC-INVNO TO APINVC-INVNO.                                   000.0005702
211000       MOVE INVC-DATE TO APINVC-DATE.                                     000.0005762
212000       MOVE INVC-DUE-DATE TO APINVC-DUE-DATE.                             000.0005822
213000       MOVE INVC-AMOUNT TO APINVC-AMOUNT.                                 000.0005882
214000       IF INVC-AMOUNT < 0 MOVE "-" TO APINVC-SAMOUNT.                     000.0005919
215000       MOVE INVC-PAYABLE TO APINVC-PAYABLE.                               000.0005995
216000       IF INVC-PAYABLE < 0 MOVE "-" TO APINVC-SPAYABLE.                   000.0006032
217000       MOVE INVC-DESC TO APINVC-DESC.                                     000.0006108
218000       MOVE 88 TO ODESY-LENGTH.                                           000.0006168
219000       GO TO WRITE-REMOTE.                                                000.0006190
220000   LAST-INVOICE.                                                          000.0006207
221000       MOVE 0 TO THIS-INV(ODESY-STATION).                                 000.0006207
222000 *     CHANGE FORMAT NAME                                                 000.0006226
223000       MOVE "VEND" TO ODESY-FORMAT.                                       000.0006226
224000       GO TO WRITE-REMOTE.                                                000.0006272
225000   END-OF-JOB.                                                            000.0006289
```

Figure 9-17. COBOL Compile Listing (4 of 5).

```
  C  O  D  E     D  I  C  T  I  O  N  A  R  Y
      BYTE LENGTH   CODEFILE RELATIVE DISK ADR
      ...........   ..........................
000 00000787        000009
    00000787        TOTAL CODE

  D  A  T  A     D  I  C  T  I  O  N  A  R  Y
      BYTE LENGTH   CODEFILE RELATIVE DISK ADR
      ...........   ..........................
000 00000730        000003

  S - M  A  C  H  I  N  E     P  A  R  A  M  E  T  E  R  S     (S  C  R  A  T  C  H  P  A  D)

  LENB - 0, SEGB - 0, DISPB - 11, COPXB - 5, COPB - 23, D.E.F - 430, BDISPB - 13
  BASE RELATIVE ADDRESSES -
    DATA-SEGMENT-0 - 688,COP-TABLE - 41

  P  R  O  G  R  A  M     P  A  R  A  M  E  T  E  R     B  L  O  C  K

  FIRST-EXECUTABLE-INSTRUCTION 0,0
  INTERPRETER-NAME             COBOL      /INTERP
  STATIC-CORE                  5840 BITS
  DYNAMIC-CORE                 0 BITS
  DATA DICTIONARY STARTS AT CODEFILE SEGMENT 2, 1 ENTRY
  CODE DICTIONARY STARTS AT CODEFILE SEGMENT 8, 1 ENTRY
  FILE PARAMETER BLOCKS START AT CODEFILE SEGMENT 14, 3 ENTRIES

***** COMPILATION COMPLETE
ELAPSED TIME - 04 MINUTES, 25 SECONDS, WHICH IS 265 SECONDS
PROCESSOR TIME - 26 SECONDS, WHICH IS 26 SECONDS
PROGRAM REQUIRES 17 DISK SEGMENTS OF 180 BYTES EACH.
MEMORY REQUIREMENTS
  0000787 BYTES - LARGEST CODE SEGMENT
  0000730 BYTES - BASE-TO-LIMIT AREA
  0000325 BYTES - DICTIONARIES AND RUN STRUCTURE
  0024162 BYTES - FILE BUFFERS & FILE INFO AREAS - INCLUDES 129 BYTES (+72 TO 540 IF DISK) FOR EACH FILE
  0026004 BYTES - ESTIMATED MEMORY REQUIRED TO RUN IF ALL FILES OPEN
226 SYMBOLIC RECORDS COMPILED AT 51.120 RECORDS PER ELAPSED MINUTE OR 521.520 RECORDS PER PROCESSOR MINUTE.
```

Figure 9-18. COBOL Compile Listing (5 of 5).

```
B U R R O U G H S  B1800/B1700 R P G   C O M P I L E R ,   LEVEL -  MARK IX.1

 P R O G R A M   N A M E     ODY20DEM00/APINQ.R
DATE COMPILED--/01/27/81     TIME COMPILED--13 47     WITH COMPILER-- DATED  08/21/80 11 18


01000F*  1         2         3         4         5         6         7         8
01100F*890123456789012345678901234567890123456789012345678901234567890123456789
01200F*
01300F*     ************************************************************
01400F*     *                                                          *
01500F*     *              A P    I N Q U I R Y    P R O G R A M       *
01600F*     *                                                          *
01700F*     ************************************************************
01800F*
01900F*                    U S E   O F   I N D I C A T O R S
02000F*
02100F*          01  DUMMY PRIMARY FILE RECORD
02200F*          02  VENDOR MASTER FILE RECORD
02300F*          03  INVOICE FILE RECORD
02400F*          04  STATIONS (ODESY) FILE RECORD
02500F*
02600F*          10  GROUP = "AP" FORMAT = "VEND"
02700F*          11  GROUP = "AP" FORMAT = "NAME" MSG.TYPE = 02  SUPPLY
02800F*          12  GROUP = "AP" FORMAT = "NAME" MSG.TYPE = 03  CHECK
02900F*          13  GROUP = "AP" FORMAT = "INVC"
03000F*          14  GROUP = "AP"
03100F*
03200F*          20  INVALID KEY
03300F*          30  SUPPLY VENDOR DATA
03400F*          31  NEG SIGN TO APNAME.PAYBAL
03500F*          40  EOF VENDMSTR
03600F*          50  EOF INVOICE
03700F*          51  SUPPLY INVOICE DETAILS
03800F*          52  NEG SIGN AMOUNT
03900F*          53  NEG SIGN PAYBAL
04000F*          60  SEND TO TERMINAL ODESY HEADER
04100F*          61  SEND TO TERMINAL ODESY HEADER & VENDOR REC
04200F*          62  SEND TO TERMINAL ODESY HEADER & INVOICE REC
04300F*          63  SEND TO TERMINAL ODESY HEADER & "MISSING VENDOR" MSG
04400F*
04500F*     ************************************************************
04600F*     *                                                          *
04700F*     *       F I L E    S P E C I F I C A T I O N S             *
04800F*     *                                                          *
04900F*     ************************************************************
05000F*
05100F*  1         2         3         4         5         6         7         8
05200F*890123456789012345678901234567890123456789012345678901234567890123456789
1 05300FDUMMY   IPE F1600  80             DISK
2 05400FVENDMSTRIC  F3680 184             DISK
3 05500FINVOICE IC  F2160 108             DISK
4 05600FSTATIONSCD  F 190 190             REMOTE
```

Figure 9-19. RPG Compile Listing (1 of 6).

```
   05800E*
   05900E*     ****************************************************
   06000E*     *                                                  *
   06100E*     *     E X T E N S I O N S   S P E C I F I C A T I O N S
   06200E*     *                                                  *
   06300E*     ****************************************************
   06400E*
   06500E* STATION TABLE
5  06600E                       TVD         10  6 0
6  06700E                       NVD         10  6 0
7  06800E                       TIV         10  8 0

   06900T*
   07000T*     ****************************************************
   07100T*     *                                                  *
   07200T*     *     T E L E C O M M U N I C A T I O N S   C A R D
   07300T*     *                                                  *
   07400T*     ****************************************************
   07500T*   1         2         3         4         5         6         7         8
   07600T*8901234567890123456789012345678901234567890123456789012345678901234567890
8  07700TSTATIONS   10 10REMSTAREMLEN
```

Figure 9-20. RPG Compile Listing (2 of 6).

```
078001I*
079001I*     ************************************************************
080001I*     *                                                          *
081001I*     *            I N P U T   S P E C I F I C A T I O N S       *
082001I*     *                                                          *
083001I*     ************************************************************
084001I*
085001I*  1         2         3         4         5         6         7         8
086001I*890123456789012345678901234567890123456789012345678901234567890123456789
 9 087001IDUMMY    NS  01
10 088001I                                        1  80 DREC
11 089001IVENDMSTRNS  02
12 090001I                                        1   6 0VVNDNO
13 091001I                                        7  30 VNAME
14 092001I                                       31  54 VADDR
15 093001I                                       55  78 VADDR2
16 095001I                                       95 1030VPYBAL
17 097001I                                      139 1440VDATLA
   098001I* INVOICE FILE
18 099001IINVOICE NS  03
19 099501I                                        1   6 0IVNDNO
20 100001I                                        7  12 0ITRNNO
21 101001I                                       13  22 0IIVNNO
22 102001I                                       23  28 0IDATE
23 103001I                                       29  34 0IDUEDT
24 104001I                                       35  43 0IAMT
25 105001I                                       44  52 0IPAID
26 106001I                                       53  61 0IPYBAL
27 108001I                                       81 104 IDESC
   109001I* REMOTE FILE STATIONS
   110001I* CHECK FOR GROUP AP AND FORMAT VEND
28 111001ISTATIONSNS   10   15 CA   16 CP   19 CV
29 112001I         AND       20 CE   21 CN   22 CD
30 113001I         OR    11  01 C0   02 C2   15 CA
31 114001I         AND       16 CP   19 CN   20 CA
32 115001I         AND       21 CM   22 CE
33 116001I         OR    12  01 C0   02 C3   15 CA
34 117001I         AND       16 CP   19 CN   20 CA
35 118001I         AND       21 CM   22 CE
36 119001I         OR    13  15 CA   16 CP   19 CI
37 120001I         AND       20 CN   21 CV   22 CC
38 123001I                                        1  20 0DYTYP
39 124001I                                        3   5 0ST
40 125001I                                        6   7 0DYOPR
41 126001I                                        8  10 0DYBCH
42 127001I                                       11  14 0DYREC
43 128001I                                       15  18 0DYGRP
44 129001I                                       19  22 0DYFMT
45 130001I                                       23  26 0DYLEN
46 131001I                                       27  28 0DYERR
47 132001I                                        1  30 0DYMSG
   133001I* APVEND RECORD
48 133501I                                      125 130 0ANNXVD
49 134001I                                       31  36 0AVVNDN
50 158501I         NS  04
51 158551I                                        1 130 BS
   159001I*
```

Figure 9-21. RPG Compile Listing (3 of 6).

```
    16000C*
    16100C*     *************************************************************
    16200C*     *                                                           *
    16300C*     *   C A L C U L A T I O N   S P E C I F I C A T I O N S     *
    16400C*     *                                                           *
    16500C*     *************************************************************
    16600C*  1         2         3         4         5         6         7         8
    16700C*8901234567890123456789012345678901234567890123456789012345678901234567890
 52 16800C           LOOP      TAG
 53 16900C                     SETOF                     010203
 54 17000C                     SETOF                     041011
 55 17100C                     SETOF                     121330
 56 17200C                     SETOF                     314050
 57 17300C                     SETOF                     515253
 58 17400C                     SETOF                     606162OUTPUT INDICATORS
 59 17500C                     SETOF                     99
 60 17600C                     RECV STATIONS             99  99
 61 17700C   99                GOTO ENDIT                      IF EOF,TERMINATE
    17800C* INVALID FORMAT (GROUP NOT = "AP")
 62 17900C   04                MOVE 13       ODYTYP
 63 18000C   04                SETON                     60
 64 18100C   04                SEND STATIONS
 65 18101C   04                EXCPT
 66 18200C   04                GOTO LOOP
    18300C* APVEND-KEYDATA (GROUP = "AP" FORMAT = "VEND")
 67 18400C   10                MOVE AVVNDN   TVD,ST           LOAD STATIONS
 68 18500C   10                MOVE AVVNDN   NVD,ST           TABLE
 69 18600C   10                MOVE 0        TIV,ST
 70 18700C   10                GOTO LOOP
 71 18800C           TVD,ST    COMP 0                        20
    18900C* INVALID KEY
 72 19000C   20                MOVE 12       ODYTYP
 73 19100C   20                SETON                     60
 74 19200C   20                SEND STATIONS
 75 19201C   20                EXCPT
 76 19300C   20                GOTO LOOP
    19400C* APNAME-SUPPLY  (GROUP = "AP" FORMAT = "NAME")
 77 19500C      11   NVD,ST    COMP 0                        30SUPPLY DATA
    19600C* FIRST INVOICE
 78 19700C      11 30          MOVE TVD,ST   NVD,ST
 79 19800C      11 30          MOVE 'INVC'   ODYFMT
 80 19900C      11 30          MOVE 0        TIV,ST
 81 20000C      11 30          MOVE TIV,ST   IKEY    50        APINVC SUPPLY
 82 20100C      11 30          EXSR READI                      CALL READ INV
 83 20200C      11 30          GOTO LOOP
    20300C* END FIRST INVOICE
 84 20400C      11             MOVE NVD,ST   TVD,ST            ELSE APNAME SUP
 85 20500C      11             MOVE TVD,ST   VKEY    50
 86 20600C      11   VKEY      CHAINVENDMSTR             40    READ VENDORFILE
    20700C* IF EOF, SEND "MISSING VENDOR" MESSAGE BACK TO STATION
 87 20800C      11 40          EXSR MISSVN
 88 20900C      11 40          GOTO LOOP
 89 21900C      11             SETON                     61     SEND HEADER,
 90 22000C      11             SEND STATIONS                 99 VENDOR REC
 91 22001C      11             EXCPT
 92 22100C      11             GOTO LOOP
    22200C* END APNAME-SUPPLY PROCEDURE
    22300C* BEGIN APNAME-CHECK
 93 22400C      12             MOVE ANNXVD   NVD,ST            APNAME NEXTVEND
 94 22500C      12             MOVE 11       ODYTYP
 95 22600C      12             SETON                     60     SEND ODESY
 96 22700C      12             SEND STATIONS                 99 HEADER
 97 22701C      12             EXCPT
 98 22800C      12             GOTO LOOP
    22900C* END APNAME-CHECK PRODEDURE
    23000C* BEGIN APINVC-SUPPLY (GROUP = "AP" FORMAT = "INVC")
 99 23100C   13                MOVE TVD,ST   IKEY    50
100 23200C   13                EXSR READI                      CALL READI
101 23300C  N99                GOTO LOOP
102 23400C           ENDIT     TAG
103 23500C                     SETON                     LR
    23600C***************
    23700C* SUBROUTINES *
    23800C***************
    23900C* MISSING VENDOR
104 24000CSR         MISSVN    BEGSR
105 24400CSR                   SETON                     63     SEND ODESY
106 24500CSR                   SEND STATIONS                 99 HEADER
107 24501CSR                   EXCPT
108 24600CSR                   ENDSR
    24700C* READ INVOICE
109 24800CSR         READI     BEGSR
110 24900CSR         IKEY      ADD  1        IKEY
111 25000CSR         IKEY      CHAININVOICE              50
112 25100CSR 50                MOVE 0        TIV,ST            LAST INVOICE
113 25200CSR 50                MOVE 'VEND'   ODYFMT
114 25300CSR 50                MOVE 11       ODYTYP
115 25400CSR 50                SETON                     60
116 25500CSR 50                SEND STATIONS                 99
117 25501CSR 50                EXCPT
118 25550CSR 50                GOTO ENDRI
119 25600CSR 50      TVD,ST    COMP IVNDNO                   51
120 25650CSRN51                GOTO ENDRI
121 25700CSR 51                MOVE IKEY     TIV,ST            SUP INV DETAILS
122 27200CSR 51                SETON                     62      SEND ODESY
123 27300CSR 51                SEND STATIONS                     MSG,INV REC
124 27301CSR 51                EXCPT
125 27350CSR         ENDRI     TAG
126 27400CSR                   ENDSR
    27500C* END READ INVOICE
```

Figure 9-22. RPG Compile Listing (4 of 6).

```
      276000*
      277000*     ************************************************************
      278000*     *                                                          *
      279000*     *          O U T P U T   S P E C I F I C A T I O N S       *
      280000*     *                                                          *
      281000*     ************************************************************
      282000*
  127 283000STATIONSE         60
  128 284000                          ODYMSG     30
  129 285000                          ODYTYPX     2
  130 285500           E      61
  131 286000                          ODYMSG     30
  132 287000                                      2 '11'
  133 288000                                     26 '0100'
  134 289000                          VVNDNOX    36
  135 290000                          VNAME      60
  136 291000                          VADDR      84
  137 292000                          VADDR2    108
  138 292500                          VPYBALL   118
  139 293000                          VDATLAX   124
  140 293500           E      62
  141 294000                          ODYMSG     30
  142 295000                                      2 '11'
  143 296000                                     26 '0088'
  144 297000                          IVNDNOX    36
  145 298000                          IIRNNOX    42
  146 299000                          IIVNNOX    52
  147 300000                          IDATE_X    58
  148 301000                          IDUEDTX    64
  149 302000                          IAMT___L   74
  150 303000                          IPAID_L    84
  151 304000                          IPYBALL    94
  152 305000                          IDESC     118
  153 305500           E      63
  154 306000                          ODYMSG     30
  155 307000                                      2 '11'
  156 308000                                     26 '0100'
  157 309000                          TVD-STX    36
  158 310000                                     60 'MISSING VENDOR'

DIAGNOSTICS COMPLETE - NO SYNTAX ERRORS
```

Figure 9-23. RPG Compile Listing (5 of 6).

```
RPG COMPILER LEVEL  MARK IX.1, DATED 08/21/80

              NON-OVERLAYABLE MEMORY -    1589 BYTES
                  TOTAL FILE SPACE -  8771 BYTES -ASSUMING ALL FILES OPEN-
                 MINIMUM CODE SPACE -   698 BYTES
                                     ------
   PROGRAM DEPENDENT REQUIRED MEMORY - 11058 BYTES
            ESTIMATE OF MCP OVERHEAD -   241 BYTES
                                     ------
*** ESTIMATED MEMORY REQUIRED TO RUN -   11299 BYTES ***

OBJECT PROGRAM REQUIRES 51 DISK SEGMENTS OF 180 BYTES EACH
ELAPSED TIME 856 SECONDS
158 SOURCE CARDS COMPILED AT 11 CARDS PER MINUTE
```

Figure 9-24. RPG Compile Listing (6 of 6).

```
BURROUGHS B1800/B1700 UPL COMPILER, MARK IX.1 (10/03/80 22:46)     TUESDAY,    1/27/81, 12:19:10 PM

                         ******************************************
                         *                                        *
                         *              JOB NUMBER:               *
                         *                000777                  *
                         *                                        *
                         *             OBJECT FILE:               *
                         *          ODY20DEMOO/APINQ.U            *
                         *                                        *
                         *              CARDS FILE:               *
                         *           ODY20DEMS/APINQ.U            *
                         *                                        *
                         ******************************************

LL  NL  SEQUENCE :SOURCE IMAGE                                                                  :  PROCEDURE   SEGMENT    PATCH

 0   0  00001000 :&LIST USEDOTS SUPPRESS                                                         :
 0   0  00002000 :%**********************************************************************:
 0   0  00003000 :%*                                                                     *:
 0   0  00004000 :%*     TITLE : B1000 ODESY                                             *:
 0   0  00005000 :%*             SAMPLE USER PROGRAM - APINQ                             *:
 0   0  00006000 :%*                                                                     *:
 0   0  00007000 :%**********************************************************************:
 0   0  00008000 :%                                                                      :
 0   0  00009000 :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:
 0   0  00010000 :%                                                                     %:
 0   0  00011000 :%     PURPOSE      : TO DEMONSTRATE SOME OF THE MORE SOPHISTICATED     %:
 0   0  00012000 :%                    INQUIRY FACILITIES OFFERED BY ODESY.              %:
 0   0  00013000 :%                                                                     %:
 0   0  00014000 :%     INPUT        : ODESY.RECORD.AREA FROM THE EDIT PROGRAM,          %:
 0   0  00015000 :%                    VENDOR MASTER FILE,                               %:
 0   0  00016000 :%                    INVOICE FILE.                                     %:
 0   0  00017000 :%                                                                     %:
 0   0  00018000 :%     OUTPUT       : ODESY.RECORD.AREA                                 %:
 0   0  00019000 :%                                                                     %:
 0   0  00020000 :%     PROCESS      : READ THE STATIONS FILE FOR INPUT. IF A "VEND"     %:
 0   0  00021000 :%                    FORMAT IS READ, THEN SET UP THE VENDOR POINTERS. IF %:
 0   0  00022000 :%                    A "NAME" FORMAT IS READ, THEN EITHER SUPPLY VENDOR %:
 0   0  00023000 :%                    INFORMATION FROM THE VENDOR MASTER FILE, OR CHECK %:
 0   0  00024000 :%                    THE DATA KEYED IN DEPENDING ON THE ODESY.MSG.TYPE. %:
 0   0  00025000 :%                    IF A "INVC" FORMAT IS READ, THEN SUPPLY INFORMATION %:
 0   0  00026000 :%                    FROM THE INVOICE FILE. EDIT WILL TERMINATE THIS   %:
 0   0  00027000 :%                    PROGRAM.                                          %:
 0   0  00028000 :%                                                                     %:
 0   0  00029000 :%     ASSUMPTIONS: VENDOR MASTER FILE CONTAINS ALL VALID VENDOR        %:
 0   0  00030000 :%                    RECORDS KEYED BY VENDOR NUMBER. THE INVOICE FILE  %:
 0   0  00031000 :%                    SHOULD CONTAIN INVOICES AND THE PROGRAM IS STARTED %:
 0   0  00032000 :%                    UP BY EDIT.                                       %:
 0   0  00033000 :%                                                                     %:
 0   0  00034000 :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:
 0   0  00035000 :%                                                                      :
 0   0  00036000 :DEFINE                                                                 :
 0   0  00037000 :          DEC AS #DECIMAL#,                                            :
 0   0  00038000 :          BIN AS #BINARY#,                                             :
 0   0  00039000 :          SPACES AS # " " #,                                           :
 0   0  00040000 :          CH AS #CHARACTER#;                                           :
 0   0  00041000 :&PAGE                                                                  :
```

Figure 9-25. UPL Compile Listing (1 of 6).

```
0   0   00042000 :FILE VENDMASTER (DEVICE = DISK RANDOM, RECORDS = 184/20, LOCK,
0   0   00043000 :                AREAS = 2/500, LABEL = "VEND)MASTER");
0   0   00044000 :FILE INVOICE    (DEVICE = DISK RANDOM, RECORDS = 108/20, LOCK,
0   0   00045000 :                AREAS = 2/500, LABEL = "INVOICE");
0   0   00046000 :FILE STATIONS   (DEVICE = REMOTE);
0   0   00047000 :%
0   0   00048000 :DECLARE
0   0   00049000 :         01 VENDOR.REC.AREA,
0   0   00050000 :            03 VEND.VENDNO                CH(6),
0   0   00051000 :            03 VEND.NAME                  CH(24),
0   0   00052000 :            03 VEND.ADDR                  CH(24),
0   0   00053000 :            03 VEND.ADDR2                 CH(24),
0   0   00054000 :            03 FILLER                     CH(16),
0   0   00055000 :            03 VEND.PAYBAL                CH(9),
0   0   00056000 :            03 FILLER                     CH(35),
0   0   00057000 :            03 VEND.DATE.LA               CH(6),
0   0   00058000 :            03 FILLER                     CH(43),
0   0   00059000 :%
0   0   00060000 :         01 INVOICE.REC.AREA,
0   0   00061000 :            03 INVC.VENDNO                CH(6),
0   0   00062000 :            03 INVC.TRANNO                CH(6),
0   0   00063000 :            03 INVC.INVNO                 CH(10),
0   0   00064000 :            03 INVC.DATE                  CH(6),
0   0   00065000 :            03 INVC.DUE.DATE              CH(6),
0   0   00066000 :            03 INVC.AMOUNT                CH(9),
0   0   00067000 :            03 INVC.PAID                  CH(9),
0   0   00068000 :            03 INVC.PAYABLE               CH(9),
0   0   00069000 :            03 FILLER                     CH(19),
0   0   00070000 :            03 INVC.DESC                  CH(24),
0   0   00071000 :            03 FILLER                     CH(4),
0   0   00072000 :&PAGE
```

```
0   0   00073000 :         01 ODESY.RECORD.AREA,
0   0   00074000 :            03 ODESY.MSG.TYPE             CH(2),% NUMERIC
0   0   00075000 :            03 ODESY.STATION              CH(3),% NUM
0   0   00076000 :            03 ODESY.OPERATOR             CH(2),% NUM
0   0   00077000 :            03 ODESY.BATCH                CH(3),% NUM
0   0   00078000 :            03 ODESY.RECORD               CH(4),% NUM
0   0   00079000 :            03 ODESY.GROUP                CH(4),%
0   0   00080000 :            03 ODESY.FORMAT               CH(4),%
0   0   00081000 :            03 ODESY.LENGTH               CH(4),% NUM
0   0   00082000 :            03 ODESY.ERRORS               CH(2),% NUM
0   0   00083000 :            03 FILLER                     CH(2),%
0   0   00084000 :            03 ODESY.DATA                 CH(160),
0   0   00085000 :%
0   0   00086000 :         01 APVEND.DATA REMAPS ODESY.RECORD.AREA,
0   0   00087000 :            03 FILLER                     CH(30),
0   0   00088000 :            03 APVEND.VENDNO              CH(6),
0   0   00089000 :%
0   0   00090000 :         01 APNAME.AREA REMAPS ODESY.RECORD.AREA,
0   0   00091000 :            03 FILLER                     CH(30),
0   0   00092000 :            03 APNAME.VENDNO              CH(6),
0   0   00093000 :            03 APNAME.NAME                CH(24),
0   0   00094000 :            03 APNAME.ADDR                CH(24),
0   0   00095000 :            03 APNAME.ADDR2               CH(24),
0   0   00096000 :            03 APNAME.SPAYBAL             CH(1),
0   0   00097000 :            03 APNAME.PAYBAL              CH(9),
0   0   00098000 :            03 APNAME.DATE                CH(6),% NUM
0   0   00099000 :            03 APNAME.NEXTVEND            CH(6),% NUM
0   0   00100000 :%
0   0   00101000 :         01 APINVC.AREA REMAPS ODESY.RECORD.AREA,
0   0   00102000 :            03 FILLER                     CH(30),
0   0   00103000 :            03 APINVC.VENDNO              CH(6),% NUM
0   0   00104000 :            03 APINVC.TRANNO              CH(6),% NUM
0   0   00105000 :            03 APINVC.INVNO               CH(10),%NUM
0   0   00106000 :            03 APINVC.DATE                CH(6),% NUM
0   0   00107000 :            03 APINVC.DUE.DATE            CH(6),% NUM
0   0   00108000 :            03 APINVC.SAMOUNT             CH(1),
0   0   00109000 :            03 APINVC.AMOUNT              CH(9),% NUM
0   0   00110000 :            03 APINVC.SPAID               CH(1),
0   0   00111000 :            03 APINVC.PAID                CH(9),
0   0   00112000 :            03 APINVC.SPAYABLE            CH(1),
0   0   00113000 :            03 APINVC.PAYABLE             CH(9),
0   0   00114000 :            03 APINVC.DESC                CH(24),
0   0   00115000 :%
0   0   00116000 :         VENDOR.KEY                       FIXED,
0   0   00117000 :         INVOICE.KEY                      FIXED,
0   0   00118000 :         I                                BIT(8),
0   0   00119000 :%
0   0   00120000 :         01 STATION.TABLE (10),
0   0   00121000 :            03 THIS.VEND                  FIXED,%
0   0   00122000 :            03 NEXT.VEND                  FIXED,%
0   0   00123000 :            03 THIS.INV                   FIXED;
0   0   00124000 :&PAGE
```

Figure 9-26. UPL Compile Listing (2 of 6).

```
0  0  :PROCEDURE WRITE.REMOTE;                                                          :
1  0  :    % SEND OK MESSAGE BACK TO ODESY STATION                                        :  WRITE.REMO
1  0  :    ODESY.MSG.TYPE := "11";                                                        :  WRITE.REMO
1  0  :    WRITE STATIONS (ODESY.RECORD.AREA);                                            :  WRITE.REMO
1  0  :END WRITE.REMOTE;                                                                  :  WRITE.REMO
0  0  :%                                                                                  :
0  0  :PROCEDURE INVALID.FORMAT;                                                          :
1  0  :    % SEND INVALID FORMAT MESSAGE BACK TO ODESY STATION. ODESY GROUP IS            :  INVALID.FO
1  0  :    % EQUAL TO "AP".                                                               :  INVALID.FO
1  0  :    ODESY.MSG.TYPE := "13";                                                        :  INVALID.FO
1  0  :    WRITE STATIONS (ODESY.RECORD.AREA);                                            :  INVALID.FO
1  0  :END INVALID.FORMAT;                                                                :  INVALID.FO
0  0  :%                                                                                  :
0  0  :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:
0  0  :%      IF ALL OF A VENDOR'S INVOICES ARE LOOKED AT, THEN CHANGE THE               %:
0  0  :%      SCREEN TO THE VENDOR SCREEN SO THAT THE OPERATOR CAN SELECT                %:
0  0  :%      ANOTHER VENDOR'S INVOICES TO LOOK AT.                                      %:
0  0  :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:
0  0  :PROCEDURE LAST.INVOICE;                                                            :
1  0  :%                                                                                  :  LAST.INVOI
1  0  :    THIS.INV(BIN(ODESY.STATION)) := 0;                                             :  LAST.INVOI
1  0  :    % CHANGE FORMAT NAME                                                           :  LAST.INVOI
1  0  :    ODESY.FORMAT := "VEND";                                                        :  LAST.INVOI
1  0  :    WRITE.REMOTE;                                                                  :  LAST.INVOI
1  0  :END LAST.INVOICE;                                                                  :  LAST.INVOI
0  0  :&PAGE                                                                              :
```

```
0  0  :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:
0  0  :%      IF THERE IS AN INVOICE FOR THE VENDOR THEN TRANSFER THE INFO               %:
0  0  :%      FROM THE INVOICE TO THE ODESY STATION. OTHERWISE SWITCH TO THE             %:
0  0  :%      VENDOR SCREEN.                                                              %:
0  0  :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:
0  0  :PROCEDURE READ.INVOICE;                                                            :
1  0  :%                                                                                  :  READ.INVOI
1  0  :    DO READ.INVOICE.FILE FOREVER;                                                  :  READ.INVOI
1  1  :        BUMP INVOICE.KEY;                                                          :  READ.INVOI
1  1  :        READ INVOICE [INVOICE.KEY] (INVOICE.REC.AREA);                             :  READ.INVOI
1  1  :            ON EXCEPTION                                                           :  READ.INVOI
1  1  :                DO;                                                                :  READ.INVOI
1  2  :                    DISPLAY "READ INVOICE EXCEPTION ERROR ";                       :  READ.INVOI
1  2  :                    LAST.INVOICE;                                                  :  READ.INVOI
1  2  :                    UNDO READ.INVOICE.FILE;                                        :  READ.INVOI
1  2  :                END;                                                               :  READ.INVOI
1  2  :            ON EOF                                                                 :  READ.INVOI
1  1  :                DO;                                                                :  READ.INVOI
1  2  :                    DISPLAY "READ INVOICE EOF ERROR";                              :  READ.INVOI
1  2  :                    LAST.INVOICE;                                                  :  READ.INVOI
1  2  :                    UNDO READ.INVOICE.FILE;                                        :  READ.INVOI
1  2  :                END;                                                               :  READ.INVOI
1  2  :        IF THIS.VEND(BIN(ODESY.STATION)) = BIN(INVC.VENDNO) THEN                   :  READ.INVOI
1  1  :            DO SUPPLY.INVOICE.DETAILS;                                             :  READ.INVOI
1  2  :                THIS.INV(BIN(ODESY.STATION)) := INVOICE.KEY;                       :  READ.INVOI
1  2  :                ODESY.DATA := SPACES;                                              :  READ.INVOI
1  2  :                APINVC.VENDNO := INVC.VENDNO;                                      :  READ.INVOI
1  2  :                APINVC.TRANNO := INVC.TRANNO;                                      :  READ.INVOI
1  2  :                APINVC.INVNO := INVC.INVNO;                                        :  READ.INVOI
1  2  :                APINVC.DATE := INVC.DATE;                                          :  READ.INVOI
1  2  :                APINVC.DUE.DATE := INVC.DUE.DATE;                                  :  READ.INVOI
1  2  :                APINVC.AMOUNT := INVC.AMOUNT;                                      :  READ.INVOI
1  2  :                IF BIN(INVC.AMOUNT) < 0 THEN                                       :  READ.INVOI
1  2  :                    APINVC.SAMOUNT := "-";                                         :  READ.INVOI
1  3  :                APINVC.PAYABLE := INVC.PAYABLE;                                    :  READ.INVOI
1  2  :                IF BIN(INVC.PAYABLE) < 0 THEN                                      :  READ.INVOI
1  2  :                    APINVC.SPAYABLE := "-";                                        :  READ.INVOI
1  3  :                APINVC.DESC := INVC.DESC;                                          :  READ.INVOI
1  2  :                ODESY.LENGTH := "0088";                                            :  READ.INVOI
1  2  :                WRITE.REMOTE;                                                      :  READ.INVOI
1  2  :                UNDO READ.INVOICE.FILE;                                            :  READ.INVOI
1  2  :            END SUPPLY.INVOICE.DETAILS;                                            :  READ.INVOI
1  2  :    END READ.INVOICE.FILE;                                                         :  READ.INVOI
1  0  :END READ.INVOICE;                                                                  :  READ.INVOI
0  0  :%                                                                                  :
0  0  :PROCEDURE APINVC.SUPPLY;                                                           :
1  0  :    INVOICE.KEY := THIS.INV(BIN(ODESY.STATION));                                   :  APINVC.SUP
1  0  :    READ.INVOICE;                                                                  :  APINVC.SUP
1  0  :END APINVC.SUPPLY;                                                                 :  APINVC.SUP
0  0  :&PAGE                                                                              :
```

Figure 9-27. UPL Compile Listing (3 of 6).

```
0  0  :PROCEDURE MISSING.VENDOR;                                                    :
1  0  :    % SEND MISSING VENDOR MESSAGE TO ODESY TERMINAL BECAUSE THE              :  MISSING.VE
1  0  :    % VENDOR NUMBER WAS NOT FOUND IN THE VENDOR MASTER FILE.                 :  MISSING.VE
1  0  :%                                                                            :  MISSING.VE
1  0  :    ODESY.DATA := SPACES;                                                    :  MISSING.VE
1  0  :    APNAME.VENDNO := CONVERT(THIS.VEND(BIN(ODESY.STATION)),CHARACTER,4);     :  MISSING.VE
1  0  :    APNAME.NAME := "MISSING VENDOR";                                         :  MISSING.VE
1  0  :    ODESY.LENGTH :="0100";                                                   :  MISSING.VE
1  0  :    WRITE.REMOTE;                                                            :  MISSING.VE
1  0  :END MISSING.VENDOR;                                                          :  MISSING.VE
0  0  :%                                                                            :
0  0  :PROCEDURE APNAME.FIRST.INVOICE;                                              :
1  0  :    NEXT.VEND(BIN(ODESY.STATION)) := THIS.VEND(BIN(ODESY.STATION));          :  APNAME.FIR
1  0  :    % CHANGE FORMAT NAME                                                     :  APNAME.FIR
1  0  :    ODESY.FORMAT := "INVC";                                                  :  APNAME.FIR
1  0  :    THIS.INV(BIN(ODESY.STATION)) := 0;               % CHECK ME              :  APNAME.FIR
1  0  :    APINVC.SUPPLY;                                                           :  APNAME.FIR
1  0  :END APNAME.FIRST.INVOICE;                                                    :  APNAME.FIR
0  0  :%                                                                            :
0  0  :PROCEDURE APNAME.CHECK;                                                      :
1  0  :    NEXT.VEND(BIN(ODESY.STATION)) := BIN(APNAME.NEXTVEND);                   :  APNAME.CHE
1  0  :    WRITE.REMOTE;                                                            :  APNAME.CHE
1  0  :END APNAME.CHECK;                                                            :  APNAME.CHE
0  0  :%                                                                            :
0  0  :PROCEDURE APNAME.SUPPLY;                                                     :
1  0  :    DO APNAME.LOOP;                                                          :  APNAME.SUP
1  1  :        IF NEXT.VEND(BIN(ODESY.STATION)) = 0 THEN                            :  APNAME.SUP
1  1  :            APNAME.FIRST.INVOICE;                                            :  APNAME.SUP
1  2  :        ELSE                                                                 :  APNAME.SUP
1  1  :              DO SUPPLY.NEXT.VENDOR.DETAILS;                                 :  APNAME.SUP
1  2  :                 THIS.VEND(BIN(ODESY.STATION)) :=                            :  APNAME.SUP
1  2  :                                 NEXT.VEND(BIN(ODESY.STATION));              :  APNAME.SUP
1  2  :                 VENDOR.KEY := THIS.VEND(BIN(ODESY.STATION));                :  APNAME.SUP
1  2  :                 READ VENDMASTER [VENDOR.KEY] (VENDOR.REC.AREA);             :  APNAME.SUP
1  2  :                      ON EXCEPTION                                           :  APNAME.SUP
1  2  :                         DO;                                                 :  APNAME.SUP
1  3  :                               DISPLAY "APNAME SUPPLY EXCEPTION ERROR";      :  APNAME.SUP
1  3  :                               MISSING.VENDOR;                               :  APNAME.SUP
1  3  :                               UNDO APNAME.LOOP;                             :  APNAME.SUP
1  3  :                         END;                                                :  APNAME.SUP
1  3  :                      ON EOF                                                 :  APNAME.SUP
1  2  :                         DO;                                                 :  APNAME.SUP
1  3  :                               DISPLAY "APNAME SUPPLY EOF ERROR";            :  APNAME.SUP
1  3  :                               MISSING.VENDOR;                               :  APNAME.SUP
1  3  :                               UNDO APNAME.LOOP;                             :  APNAME.SUP
1  3  :                         END;                                                :  APNAME.SUP
1  3  :                 IF VEND.VENDNO = " " THEN UNDO APNAME.LOOP;% NO RECORD      :  APNAME.SUP
1  2  :                 ODESY.DATA := SPACES;                                       :  APNAME.SUP
1  2  :                 APNAME.VENDNO := VEND.VENDNO;                               :  APNAME.SUP
1  2  :                 APNAME.NAME := VEND.NAME;                                   :  APNAME.SUP
1  2  :                 APNAME.ADDR := VEND.ADDR;                                   :  APNAME.SUP
1  2  :                 APNAME.ADDR2 := VEND.ADDR2;                                 :  APNAME.SUP
1  2  :                 APNAME.PAYBAL := VEND.PAYBAL;                               :  APNAME.SUP
1  2  :                 IF BIN(VEND.PAYBAL) < 0 THEN                                :  APNAME.SUP
1  2  :                    APNAME.SPAYBAL := "-";                                   :  APNAME.SUP
1  3  :                 APNAME.DATE := VEND.DATE.LA;                                :  APNAME.SUP
1  2  :                 ODESY.LENGTH := "0100";                                     :  APNAME.SUP
1  2  :                 WRITE.REMOTE;                                               :  APNAME.SUP
1  2  :                 UNDO APNAME.LOOP;                                           :  APNAME.SUP
1  2  :              END SUPPLY.NEXT.VENDOR.DETAILS;                                :  APNAME.SUP
1  2  :    END APNAME.LOOP;                                                         :  APNAME.SUP
1  0  :END APNAME.SUPPLY;                                                           :  APNAME.SUP
0  0  :&PAGE                                                                        :
```

Figure 9-28. UPL Compile Listing (4 of 6).

```
0  0  :PROCEDURE INVALID.KEY;                                                        :
1  0  :    ODESY.MSG.TYPE := "12";                                                   :  INVALID.KE
1  0  :    WRITE STATIONS (ODESY.RECORD.AREA);                                       :  INVALID.KE
1  0  :END INVALID.KEY;                                                              :  INVALID.KE
0  0  :%                                                                             :
0  0  :PROCEDURE APVEND.KEYDATA;                                                     :
1  0  :    THIS.VEND(BIN(ODESY.STATION)) := BIN(APVEND.VENDNO);                      :  APVEND.KEY
1  0  :    NEXT.VEND(BIN(ODESY.STATION)) := BIN(APVEND.VENDNO);                      :  APVEND.KEY
1  0  :    THIS.INV(BIN(ODESY.STATION)) := 0;                                        :  APVEND.KEY
1  0  :END APVEND.KEYDATA;                                                           :  APVEND.KEY
0  0  :%                                                                             :
0  0  :PROCEDURE AP.CHECK;                                                           :
1  0  :    % DETERMINE WHAT TYPE OF PROCESSING NEEDS TO BE DONE                      :  AP.CHECK
1  0  :    IF ODESY.FORMAT = "VEND" THEN                                             :  AP.CHECK
1  0  :        APVEND.KEYDATA;                                                       :  AP.CHECK
1  1  :    ELSE IF THIS.VEND(BIN(ODESY.STATION)) = 0 THEN                            :  AP.CHECK
1  1  :        INVALID.KEY;                                                          :  AP.CHECK
1  2  :    ELSE IF ODESY.FORMAT = "NAME" AND ODESY.MSG.TYPE = "02" THEN              :  AP.CHECK
1  2  :        % SUPPLY DATA                                                         :  AP.CHECK
1  2  :        APNAME.SUPPLY;                                                        :  AP.CHECK
1  3  :    ELSE IF ODESY.FORMAT = "NAME" AND ODESY.MSG.TYPE = "03" THEN              :  AP.CHECK
1  3  :        % CHECK DATA                                                          :  AP.CHECK
1  3  :        APNAME.CHECK;                                                         :  AP.CHECK
1  4  :    ELSE IF ODESY.FORMAT = "INVC" THEN                                        :  AP.CHECK
1  4  :        APINVC.SUPPLY;                                                        :  AP.CHECK
1  5  :    ELSE APVEND.KEYDATA;                                                      :  AP.CHECK
1  5  :END AP.CHECK;                                                                 :  AP.CHECK
0  0  :&PAGE                                                                         :
```

```
0  0  :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%:
0  0  :% EXECUTABLE CODE                                                             :
0  0  :OPEN VENDMASTER INPUT,OUTPUT,LOCK;                                            :
0  0  :OPEN INVOICE INPUT,OUTPUT,LOCK;                                               :
0  0  :OPEN STATIONS INPUT,OUTPUT;                                                   :
0  0  :I := 0;                                                                       :
0  0  :DO FOREVER;                                                                   :
0  1  :    THIS.VEND(I) := 0;                                                        :
0  1  :    NEXT.VEND(I) := 0;                                                        :
0  1  :    THIS.INV(I) := 0;                                                         :
0  1  :    IF ((BUMP I) > 9) THEN UNDO;                                              :
0  1  :END;                                                                          :
0  0  :%                                                                             :
0  0  :DO MAIN.LOOP FOREVER;                                                         :
0  1  :    READ STATIONS (ODESY.RECORD.AREA);                                        :
0  1  :        ON EOF UNDO MAIN.LOOP;                                                :
0  2  :    IF ODESY.GROUP = "AP" THEN                                                :
0  1  :        AP.CHECK;                                                             :
0  2  :    ELSE                                                                      :
0  1  :        INVALID.FORMAT;                                                       :
0  2  :END MAIN.LOOP;                                                                :
0  0  :FINI;                                                                         :

***** COMPILATION COMPLETE
```

Figure 9-29. UPL Compile Listing (5 of 6).

```
COMPILE STATISTICS:

     NUMBER OF ERRORS DETECTED:     0
     NUMBER OF WARNINGS DETECTED:     0
     NUMBER OF CARDS SCANNED:      305
     NUMBER OF TOKENS SCANNED:     1204
     LL ZERO NAME STACK ENTRIES:     63

PROGRAM STATISTICS

     CORE REQUIRED TO RUN:          2786 BYTES      (  22288 BITS)
     NUMBER OF SEGMENTS:            2
     SIZE OF LARGEST SEGMENT:       1280 BYTES      (  10236 BITS)
     TOTAL SEGMENT SIZE:            1280 BYTES      (  10236 BITS)
     DISK SIZE:                    19 SEGMENTS

RUN STATISTICS:

     NAME STACK SIZE:                97 ENTRIES
     CONTROL STACK SIZE:             15 ENTRIES
     PROGRAM POINTER STACK SIZE:     25 ENTRIES
     EVALUATION STACK SIZE:          20 ENTRIES
     VALUE STACK SIZE:               14784 BITS
     PROGRAM STATIC CORE:             2756 BYTES      (  22048 BITS)
     PROGRAM DYNAMIC CORE:               0 BYTES      (      0 BITS)

COMPILE TIMES:

     ELAPSED TIME:    0:02:09.9
     PROCESSOR TIME:   26.1 SECONDS.
```

Figure 9-30. UPL Compile Listing (6 of 6).

SECTION 10

SYSTEM MAINTENANCE

In the course of system operation, it may be necessary to perform various maintenance and recovery functions specifically for ODESY. Procedures for control of the system and recovery of files comprise this section.

DATA CONTROL

Various data control procedures help to ensure the proper use of ODESY. The control of data may be centralized or decentralized, depending on your business.

It is the responsibility of data control to ensure that:

1. Data passes through ODESY once.
2. Batches are extracted when they are ready.
3. Batches are deleted when they are no longer required.
4. Batch numbers are allocated logically, for example, by assigning certain ranges of numbers to certain departments.
5. All checks have been performed before the batches are extracted.
6. Regular backup is taken of files.

BACKUP

You should have a well-organized system for taking backup of the three ODESY master files on disk. Take backup on a regular basis, such as once or twice a day, depending on the volume of data being entered. Copy the files to magnetic tape or removable disk pack and use a cycle of at least three tapes or packs. It is imperative that you backup the System and Tank files at the same time when the files are not in use by an ODESY program because the time and date in each file must match.

SECURITY

System security is implemented at your discretion. A small data entry operation with minimal security requirements may operate without the restrictions of tight security, but a larger organization which needs to prevent unauthorized access to the sytem or files may use the full range of security features.

PASSWORDS

Each operator is assigned a number and password which must be entered during log-in to obtain access to ODESY data. You may impose a more rigorous system of access control through a supervisory MCS.

ACCESS LEVELS

Security levels enable you to restrict access to certain formats. Each operator is given a minimum and a maximum format level from 0 to 99 and each format is given a security level. If you do not require this feature, give all operators and formats identical security levels.

OPERATOR STATUS

A supervisor is an operator with higher status, specified by setting the supervisor status option when defining the operator to ODESY. You may enforce greater control by restricting certain commands to supervisors. These are the ACCEPT, CONTROL, DELETE, INSERT, MODIFY, SCHEDULE, TERMINATE, and ZIP commands, and the ALL form of the STATUS command.

CONTROL STATIONS

A control station is an ODESY station with control status. The control stations enable you to use the TERMINATE and ZIP commands. Usually a supervisor operates a control station, which restricts the use of the TERMINATE and ZIP commands even further. These two commands are treated differently because they affect execution of the entire system, so extra security is provided to prevent their unauthorized use.

Create a permanent control station by setting the control status option when defining the station to ODESY. Create a temporary control station through the CONTROL command. The station retains control status only for the duration of the current session.

USERCODE SECURITY

If usercode security is generally used on the system, it may be necessary for the Extraction program to create secured files. To do this, EXTRACT must be run under a privileged usercode, and the names of the secured files, with their usercodes, must be specified to FORMAINT. In this way, EXTRACT is able to create secured files under several different usercodes in the same run.

If EXTRACT is run under a usercode, it can create both secured and non-secured files. If EXTRACT is run without a usercode, it can create only non-secured files. Files under a public usercode are regarded as non-secured files, since EXTRACT does not need to be run under a usercode to create such files.

However, it should be noted that if the Extraction program is run under a usercode while the Data Entry program is up, EDIT should run under the same usercode. There are also implications from using usercode security that affect the Format Maintenance program. These implications are discussed in the section "Specification of Secured Files to FORMAINT," which follows.

Full documentation on the file security system is provided in the revised software operational guide for MCP VI.1

ODESY USERCODE

If you wish to have usercode security, we recommend you create a special usercode, such as ODESY/ODESY, to use with all of the ODESY programs. Note the following points about this special usercode:

1. It must be a privileged usercode, so that files under all usercodes can be accessed.

2. If the usercode has a default pack associated with it, ODESY programs executed under the usercode will look for their ODESYDATA files on the default pack.

   In addition, if files are specified without a usercode or pack name, the Extraction Program will create files on the default pack rather than on the system pack. Similarly, FORMAINT will create and look for dump files on the default pack.

3. It must be a public usercode. Otherwise, non-secured files will be created as private files.

4. The maximum priority for the usercode should not be less than the normal scheduled priority for EXTRACT. The default value of 7 is normally adequate.

The following options for SYSTEM/MAKEUSER should create a suitable usercode:

```
US=ODESY PW=ODESY PUBLIC *PRIV
```

SPECIFICATION OF SECURED FILES TO FORMAINT

A secured file is identified by the existence of parentheses surrounding its family name. For example:

```
(UC)/DATA
```

The family name enclosed in parentheses, such as (UC), is referred to as a usercode.

When specifying a secured file on the group header screen, the usercode must be entered as the family name and enclosed in parentheses, as is shown above. If the family name contains a usercode, then the file name is mandatory.

If the usercode has a default pack associated with it, but this particular file is to be written to the system disk, then the family name should should begin with an asterisk:

*(UC)/DATA

The following table shows the various valid alternatives for naming secured and non-secured files. It is assumed that EXTRACT is to be run under a privileged, public usercode. This usercode is ODESY/ODESY. ODESY/ODESY is shown in this table to run with both no default pack, and default pack D. The usercode UC/PASS has a default pack P.

Pack	Family	File	File Created With No Default Pack	File Created With Default Pack D
	B		(ODESY)/B	D/(ODESY)/B
A	B		A/(ODESY)/B	A/(ODESY)/B
	B	C	B/C	D/B/C
A	B	C	A/B/C	A/B/C
	(UC)	C	P/(UC)/C	P/(UC)/C
	*(UC)	C	(UC)/C	(UC)/C
A	(UC)	C	A/(UC)/C	A/(UC)/C

EXECUTION OF ODESY PROGRAMS

To avoid file compatibility problems, always run all ODESY programs under the same usercode. Otherwise, for example, the Format Maintenance Program may not be able to load from a dump file it created during the previous run.

If EXTRACT is run on-line with EDIT, it must be executed under the same usercode. This will occur automatically if you execute EXTRACT at an ODESY terminal using the ZIP command. If EXTRACT is executed from the SPO, enter:

<mix # of EDIT> AX ZIP EX ODESY/EXTRACT

USER INTERFACE PROGRAMS

If EDIT runs under a user code, then all user programs it zip executes will run under the same user code, and will use the same default pack. Thus, user programs must be given a file name as well as a family name. (The family name may not be a user code.) The default pack on which the programs will reside must also be specified on the format header screen.

RECOVERY

ODESY provides comprehensive recovery and restart facilities to cope with hardware or software failure. These facilities are an integral part of the system and are automatically brought into use when the need arises. After a hardware or software failure, it is usually sufficient to re-execute ODESY. During normal startup, ODESY recognizes that the system has terminated abnormally and invokes a recovery procedure to restore it to the state it was in prior to termination.

Recovery procedures are used by the Edit, Extraction, and Format Maintenance Programs. However, the recovery procedures assume that the three ODESY master files on disk have not been corrupted. If disk file corruption has occurred, ODESY should be able to provide a warning. It is unlikely that all of the ODESY record pointers and totals will match if the files are corrupted. If this should happen, load the most recent backup copy of the files.

EDIT PROGRAM

While the Edit program is executing, it maintains a record of the current action being processed. It also maintains a list of active stations, operators, and batches. Recovery either completes or removes the current action, closing all batches and logging out all operators. The files remain in usable condition, and the maximum amount of data in each batch which may be lost is the record currently being processed, or a maximum of one record per station.

After recovery operators must log in again, and at this point ODESY informs each operator of the batch number and record number being processed when the system failed,

helping the operator to resume work at the point of interruption.

EXTRACTION PROGRAM

If the system fails while the Extraction Program is executing, operators may notice a batch permanently "in use at station 0." A batch is marked in use while the Extraction Program is extracting or deleting it to prevent an operator from accidentally opening the batch and making changes to it at the same time the Extraction Program is reading and updating the data records. Processing of the batch marked in use will be completed during the next execution of EXTRACT.

The Extraction Program first tries to complete the extraction or deletion of the batch processed previously. Execution continues by processing any other scheduled batches. The Extraction Report shows the batches processed.

The recovery procedure of the Extraction Program is independent of that of the Edit Program, so it is not necessary to restart both programs immediately if both were executing when the system failed.

FORMAT MAINTENANCE PROGRAM

The Format Maintenance Program does not have a complex recovery system because it is unnecessary. Files are updated only during a SAVE, LOAD, or REMOVE command and take only a small percentage of the total execution time. If the system terminates abnormally at any other time, files do not require recovery. During hardware or software failure, restart the Format Maintenance Program and continue working. If a SAVE, LOAD, or REMOVE command is being processed when a failure occurs, ODESY provides a warning.

FILE INITIALIZATION

In certain cases, usually involving the corruption of a data file, you may need to create all new, empty ODESYDATA files. This procedure should be used only as a last resort, since it removes all tank file data. First, try, for example, a squash on the corrupt file or obtain three new data files from your release media. If all else fails, however, the following steps should be carefully followed.

1. Create, or obtain from backup, an ODESYSETUP/INPUT file containing a complete set of system maintenance requests. If the ODESYSETUP/INPUT file is not complete, execution of the following procedure will produce an incomplete, possibly inoperable system file. You can use the ODESYSETUP/INPUT file from the ODESY release media, if available, and reconfigure your system later.

2. Obtain a dump file of all formats, including the SYS group. Otherwise, user formats, and possibly the SYS group, will have to be re-inserted into your formats file later by being rekeyed.

3. As a precaution, change the names of your current ODESYDATA files and retain them as backup until you are certain the new files are usable.

4. Change the first record in ODESYSETUP/INPUT (the CONI record) by using CANDE or a similar means, in the following way:

   a. Replace the letters "CONI" with "INIT".

   b. Leave all of the rest of the data in the first record as is, and do not change any other records in the file.

5. Execute ODESY/FILEMAINT using the file containing the "INIT" record as the input file.

6. When FILEMAINT goes to end-of-job, run the Format Maintenance program long enough to load your formats. The SYS group of formats may be loaded from the dump file ODESYSETUP/FORDUMP on your release media, if available.

7. To obtain a print of your entire current system configuration, enter at the ODT::

```
EX     ODESY/FILEMAINT     FILE/INPUT     NAME
ODESYSETUPT/PRINT
```

If your station is not declared in the system file, insert it. (See Section 2.)

8. Bring up the Data Entry program, and re-enter batch 999.

   a. /OPEN 999 ENTER SYS

   b. //CONI

   c. Continue until your configuration, station, and operator records, at least, are entered.

> NOTE
>
> It is only necessary to enter the system maintenance requests that are used, such as the configuration, operator, or station records. If the text and command records, for example, are never changed, these need not be entered into batch 999.

9. Remove from disk the ODESYSETUP/INPUT file that contains the INIT record.

APPENDIX A

FORMAT MAINTENANCE PROGRAM COMMANDS

The Format Maintenance Program commands are used to manipulate the screen format options in ODESY, and this appendix explains how and when to use each command. Some control commands which allow you to manage system operation to a small degree are also included.

In this appendix, the commands are presented in diagrams composed of words and arrows. These diagrams show exactly what to enter to perform a task. To execute a command, trace any path along the direction of the arrows, and enter the words and symbols exactly as they appear along the path. Required words are shown in capital letters. Variables which you supply are shown within angle brackets (< >).

All ODESY commands must be preceded by a slash (/); however, it is necessary to enter only the first three letters of the command for ODESY to recognize it. The command and all parameters must be separated by at least one blank space.

NOTE

> To send a message to the MCP, use a question mark (?) rather than a slash (/) followed by a control string.

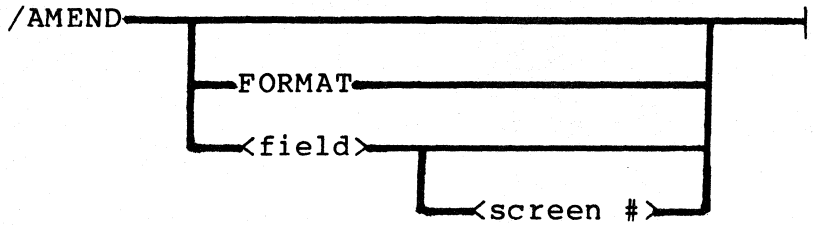
To create, list, delete, and modify screen formats, enter the appropriate commands after the program has been initiated. To enter a command:

1. Type <command>.
2. Press the transmit key (XMT).

A summary of valid commands for the Format Maintenance Program and an explanation of each command follows. When a command has more than one form, each option is explained.

AMEND

The AMEND command is used to make changes to format data in the work area.



Use this command to update an existing group header, format header, or field. Obtain the group header or format header from the Formats file before using the AMEND command.

The AMEND command has three options:

1. /AMEND

   This option changes a group header.

2. /AMEND FORMAT

   This option changes a format header.

3. /AMEND <field>

   /AMEND <field> <screen #>

   This option can be used in two forms. The first form starts a cycle in which all of the field details screens for the given field are displayed, followed by the screens for the next field, and so on. This continues until there are no more fields in the format or another command is entered.

   NOTE

   All of the screens for the field being changed, including all value-check constant screens, must be transmitted before the field is amended.

The second option permits a change to be made to one field details screen without having to "page through" all of the screens for the given field.

For example, if a field's output position needs to be changed, the only screen you need is field details 3. Enter "/AMEND <field sequence #> 3" to change that screen alone. If field details 2 or 3 is needed and a constant, range, or value check is specified, the constant screens involved are displayed immediately following the field details screen. All of these screens must be transmitted before the AMEND takes affect.

Note that an error returned for a change made on one screen may be caused by a specification appearing on another screen. For example, a change in field type from alphanumeric to numeric on the field details 2 screen may be rejected because the field length, specified on field details 1, is too large for a numeric field.

BYE

The BYE command ends the Format Maintenance Program and logs the operator out of the system.

```
/BYE
```

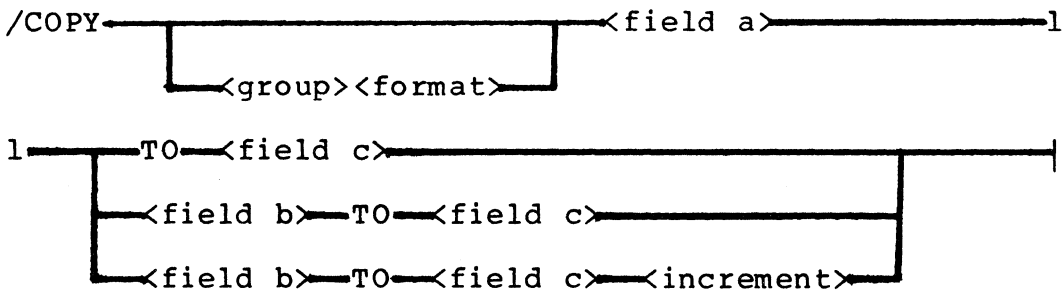
Use this command to end a format maintenance session.

Unsaved format data in the workarea must be saved or deleted before using the BYE command. The data can be deleted, or transferred to the Formats file using the SAVE command or to a Formats Dump file using the DUMP command. The Format Maintenanceprogram may also be terminated by entering the following at the ODT:

```
<mix no>AXBYE
```

COPY

The COPY command duplicates a field, or range of fields, from a format in the Formats file or from the current format in the workarea and places the new field in the workarea.



Use this command to copy a single field or group of fields from another part of the workarea or from another format. A group of fields with the same set of format options often appears several times in a format. The COPY command allows unlimited duplication of fields.

In copying fields from another format, the format specified must be in the Formats file.

The COPY command has six options:

1. /COPY <field a> TO <field c>

   This option copies a single field specified as field a to the destination indicated by field c.

2. /COPY <field a> <field b> TO <field c>

   This option copies a range of fields specified as field a through field b to the destination indicated by field c and continuing by default increments of 10.

3. /COPY <field a> <field b> TO <field c> <increment>

   This option copies a range of fields specified as field a through field b to the destination beginning at the point indicated by field c and continuing through the sequence number determined by the increment. The second field will be

numbered as field c plus the increment, the third field as field c plus two times the increment, and so on. The default increment is 10.

4. /COPY <group> <format> <field a> TO <field c>

   This option copies a single field from the format specified as field a to the destination indicated by field c.

5. /COPY <group> <format> <field a> <field b> TO <field c>

   This option copies a range of fields from the format specified as field a through field b to the destination indicated by field c and continuing by default increments of 10.

6. /COPY <group> <format> <field a> <field b> TO <field c> <increment>

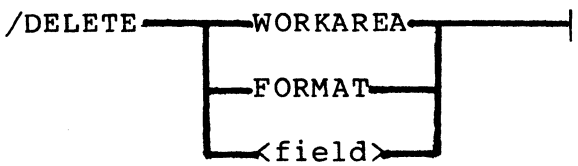
   This option copies a range of fields from the format specified as field a through field b to the destination beginning at the point indicated by field c and continuing through the sequence number determined by the increment. The second field will be numbered as field c plus the increment, the third field as field c plus two times the increment, and so on. The default increment is 10.

If copying a range of fields would result in duplicate sequence numbers or a number greater than 999, the copy is not performed and message 152 SEQUENCE NUMBERS WILL NOT FIT is displayed.

The COPY command is distinguished from the MOVE command as follows: if field 10 is copied to field 55, two identical fields, 10 and 55, will exist after the copy. If field 10 is moved to field 55, only field 55 will exist after the move.

DELETE

The DELETE command deletes format data from the workarea.



Use this command to delete all or part of the data in the workarea. Individual fields or the whole format, with or without the group header, can be deleted from the workarea.

NOTE

> Distinguish the DELETE command, which clears the work area, from the REMOVE command, which permanently removes a format or group of formats from the Formats file itself.

If you attempt to use the BYE or INSERT commands with unsaved data in the workarea, message 134 SAVE OR DELETE WORKAREA will be displayed as a warning.

The DELETE command has three options:

1. /DELETE WORKAREA

   This option deletes the entire workarea and is often used when there is data in the workarea which has been changed but is not to be saved.

2. /DELETE FORMAT

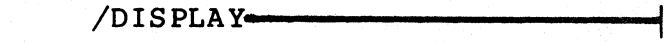
   With this option all format data other than the group header is deleted from the workarea.

3. /DELETE <field>

   This option deletes a single field from the workarea.

DISPLAY

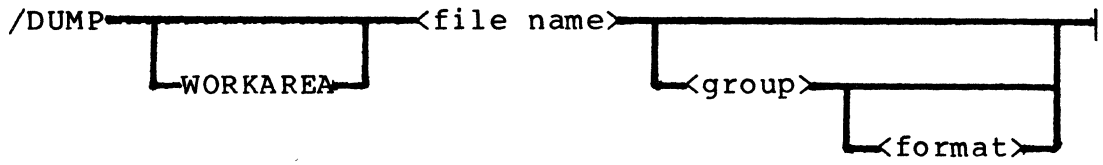
The DISPLAY command is used to view the screen layout of a format residing in the workarea.



Use this command to check the alignment of fields and field descriptions, but make changes through the AMEND command. The format must be in the workarea before using the DISPLAY command.

DUMP

The DUMP command moves format data from the Formats file or workarea to a Format Dump file.



Use this command to transfer formats to a secondary file known as a Formats Dump file.

Old or seldom-used format versions can be stored in the Dump file. It also can be used to transfer format data between different copies of the Formats file or between small and medium systems ODESY, or as temporary storage when format maintenance takes place at the same time as data entry (refer to the SAVE command).

If a Dump file with the file name specified already exists, formats are added to the existing file; otherwise a new Dump file is created.

The DUMP command has five options:

1. /DUMP <file name>

   This option transfers the contents of the Formats file to a specified Dump file.

2. /DUMP <file name> <group>

   This option transfers a group of formats in the Formats file to a specified Dump file.

3. /DUMP <file name> <group> <format>

   This option transfers a single format in the Formats file to a specified Dump file.

4. /DUMP WORKAREA <file name>

   This option transfers the contents of the workarea to a specified Dump file.

5. /DUMP WORKAREA <file name> <group>

   This option transfers the group header data in the workarea to a specified Dump file.

It is possible to have two or more formats with the same name in a Dump file when formats are repeatedly dumped to an existing file. This is because a Dump file is a serial file and new data is added to the end of the file. Dump files with duplicates load correctly, but the disadvantage of large Dump files containing many duplicates is the time required to load.

GET

The GET command transfers format data from the Formats file to the workarea where it can be manipulated by workarea commands.

Use this command to obtain a group header and format from the Formats file.

The GET and SAVE commands are jointly responsible for the transfer of data between the workarea (temporary storage) and the Formats file (permanent storage). Unsaved format data in the workarea must be saved or deleted before proceeding.

The GET command has two options:

1. /GET <group>

   This option transfers the group header from the Formats file to the workarea.

2. /GET <group> <format>

   This option transfers the group header, format header, and all fields from the Formats file to the workarea.

Message 134 SAVE OR DELETE WORKAREA is displayed as a warning that there is unsaved format data in the workarea and that the data must be saved or deleted before proceeding.

HELLO

The HELLO command logs an operator on to the system for format maintenance.

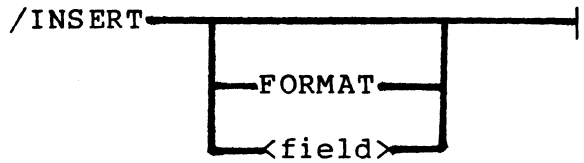
Use this command to begin a format maintenance session.

The operator number and password must refer to a valid operator defined in the System file.

Only one operator at a time may maintain formats, if an operator at another station is already logged on, message 150 OPERATOR LOGGED IN ELSEWHERE is displayed and access is denied.

INSERT

The INSERT command inserts new format data into the workarea.



Use this command to insert a new group, new format, or new field into the workarea.

The workarea can hold only one format and one group header at a time. Unsaved data in the workarea must be either deleted or transferred to the Formats file through the SAVE command or to a Formats Dump file through the DUMP command before using the INSERT command.

The INSERT command has three options:

1. /INSERT

   This option inserts a new group header followed by a format header and fields.

2. /INSERT FORMAT

   This option inserts a new format header followed by the fields of the format. The group header must be in the workarea; if a new format is being added to an existing group, obtain the group header through the GET command.

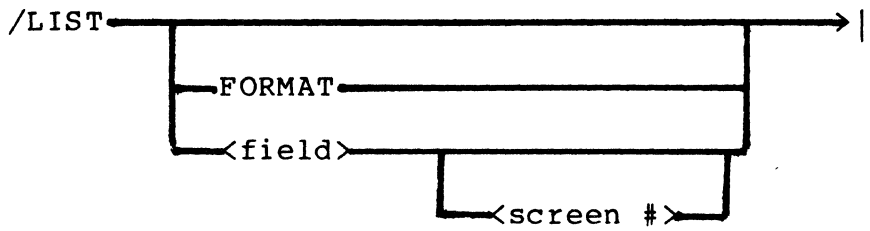
3. /INSERT <field>

   To insert a new field into the format in the workarea, specify the field sequence number. If a new field is to be inserted into an existing format in the Formats file, use the GET command to transfer the format to the workarea.

The INSERT command initiates a cycle in which the group header formats are automatically followed by the format header format and then by the field formats. The cycle repeats automatically for the insertion of more fields. Stop this cycle by entering an ODESY command.

LIST

The LIST command lists the format data contained in the workarea so that individual format options can be viewed.



Use this command to list on the screen the format options of an existing group header, format header, or field. Obtain the group header or format from the Formats file before using the LIST command.

The LIST command has four options:

1. /LIST

   This option lists a group header.

2. /LIST FORMAT

   This option lists a format header.

3. /LIST <field>

   To list all the screens for a field, specify the field sequence number.

4. /LIST <field> <screen number>

   To list a specific field details screen, specify the field sequence number followed by the screen number.

All options for this command except the fourth initiate a cycle in which the group header screens are automatically followed by the format header screen and then by the field screens. The cycle repeats automatically for the listing of other fields.

To request the next page in the cycle transmit a single blank character. Stop this cycle by using an ODESY command.

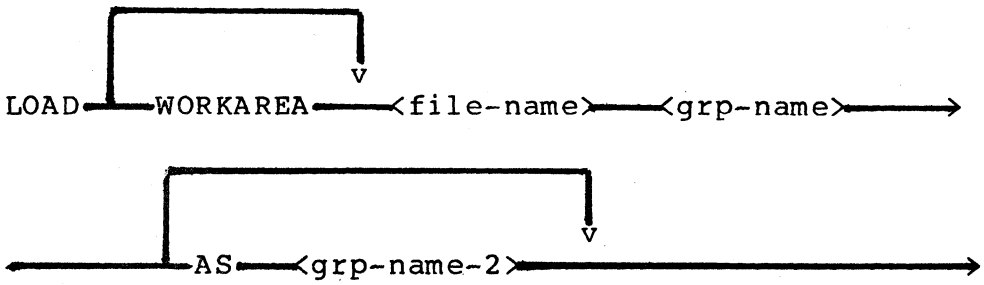
LOAD

The LOAD command transfers formats from a format dump file to the Formats file or the workarea.

This command also lets you modify the group name or the format name during the process.

This command has three options:

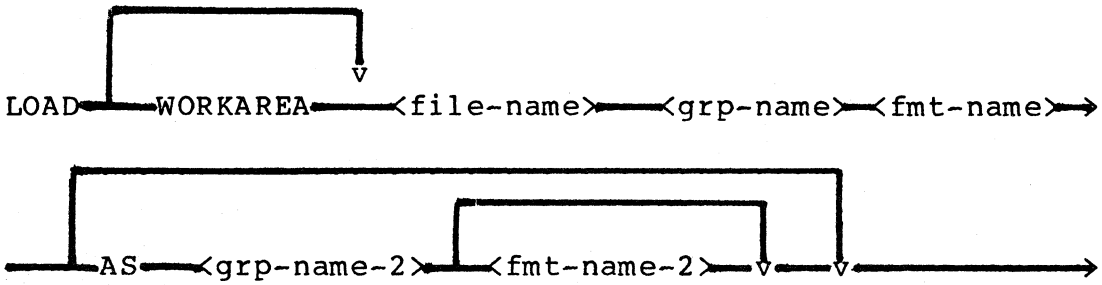
Option 1:



This option loads a group of formats to the Formats file. If WORKAREA is specified, only the group header name is loaded in the workarea.

If AS is specified, the group header name is changed to grp-name-2.

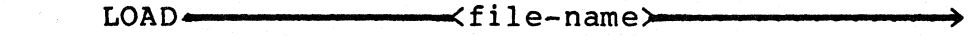
Option 2:



This option loads a single format to the format file or the work area.

If AS is specified, the group header name is changed to grp-name-2. If fmt-name-2 is also specified, the format name is changed to fmt-name-2.

Option 3:



This option loads all formats contained in the dump file into the Formats file.

Formats cannot be loaded while extraction is in progress. However, they can be loaded while the Data Entry Program is running if the following three conditions are met.

1. The On-Line Format Maintenance option has been included in the system configuration.
2. Load option 3 is not used.
3. The group to be loaded is not in use by the data entry program.

Message 144, NO FORMATS HAVE BEEN LOADED, is displayed as a warning under any of the following conditions:

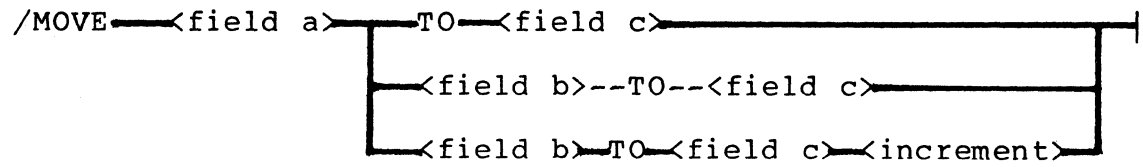
1. No formats were found for the group which was specified to be loaded.
2. There was a version mismatch.
3. Some other difficulty prevented the load from occurring.

Message 157, GROUP IN USE BY EDIT, is displayed if the group to be loaded is in use by EDIT, or no group was specified.

Message 153, FORMATS FILE IS IN USE, is displayed if extraction is in progress, or if the On-Line Format Maintenance option has been excluded in the system configuration and Edit is running.

MOVE

The MOVE command changes the order of fields within the workarea.



Use this command to move a single field or group of fields from one part of the workarea to another part. Field sequence numbers determine the order in which fields appear on the screen, so moving a field alters its position on the screen.

The MOVE command has three options:

1. /MOVE <field a> TO <field c>

   This option moves a single field specified as field a to the destination indicated by field c.

2. /MOVE <field a> <field b> TO <field c>

   This option moves a range of fields specified as field a through field b to the destination indicated by field c and continuing by default increments of 10.

3. /MOVE <field a> <field b> TO <field c> <increment>

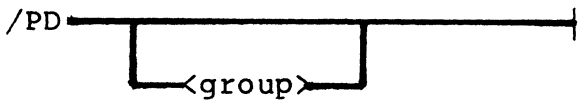
   This option moves a range of fields specified as field a through field b to the destination beginning at the point indicated by field c and continuing through the sequence number determined by the increment. The second field will be numbered as field c plus the increment, the third field as field c plus two times the increment, and so on. The default increment is 10.

If moving a range of fields would produce duplicate sequence numbers or a number greater than 999, the move is not performed and message 152 SEQUENCE NUMBERS WILL NOT FIT is displayed.

The MOVE command is distinguished from the COPY command as follows: if field 10 is moved to field 55, only one field, 55, will exist after the move. If field 10 is copied to field 55, two fields, 10 and 55, will exist after the copy.

PRINT DIRECTORY (PD)

The PRINT DIRECTORY command produces a directory listing of all group names in the Formats file or of all format names in a specific group. The listing is displayed on the requesting terminal's screen.



The PRINT DIRECTORY command has two options:

1. /PD

   This option produces a listing of all group names in the Formats file.

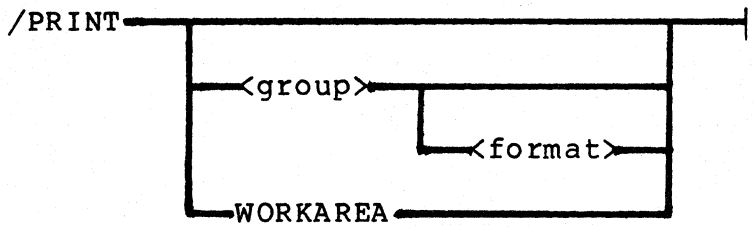
2. /PD <group>

   This option produces a listing of all format names in the group specified.

If the list of names does not fit on one screen, a field appears in the lower right corner of the screen. Transmit the cursor from this field to obtain continuation pages.

PRINT

The PRINT command produces a formatted line printer report of the data in the Formats file or workarea including a representation of the screen layout of the format.



Use this command to request a printed report of ODESY formats. Reports must be requested through this command; they are not automatically produced as data is inserted.

The PRINT command has four options:

1. /PRINT

   This option produces a report for each format in the Formats file.

2. /PRINT <group>

   This option produces a report for each format in the specified group.

3. /PRINT <group> <format>

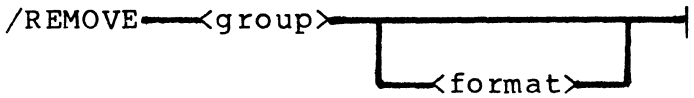
   This option produces a report showing a single format.

4. /PRINT WORKAREA

   This option produces a report showing the contents of the workarea.

REMOVE

The REMOVE command removes a specific format or group of formats from the Formats file.



The REMOVE command has two options:

1. /REMOVE <group>

   This option removes the group header and all of its formats from the Formats file.

2. /REMOVE <group> <format>

   This option removes a single format from the Formats file.

Formats cannot be removed while extraction is in progress but may be done while EDIT is running provided:

1. The On-Line Format Maintenance option has been included in the system configuration.

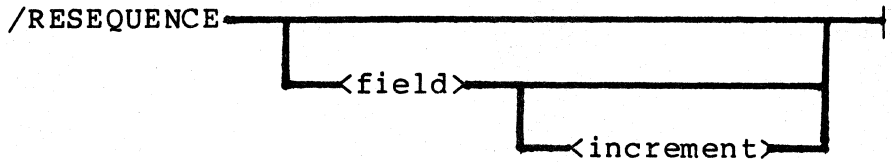
2. The group affected by the remove is not in use by EDIT.

If the Formats file is in use by EXTRACT or the above two conditions have not been met, message 153 FORMATS FILE IS IN USE, is displayed.

Distinguish the REMOVE command from the DELETE command which clears the format or group header and format out of the workarea.

RESEQUENCE

The RESEQUENCE command allocates new field sequence numbers to the fields in the workarea.



The first field in the workarea is renumbered as the base field sequence number. The second field will be numbered as the base field sequence number plus the increment, the third field as the base field sequence number plus two times the increment, and so on.

The RESEQUENCE command has three options:

1. /RESEQUENCE

   This option resequences the fields in the format workarea using 10 as a default base field sequence number and 10 as a default increment.

2. /RESEQUENCE <field>

   This option resequences the fields in the format workarea using the specified base field sequence number and 10 as a default increment.

3. /RESEQUENCE <field> <increment>

   This option resequences the fields in the format workarea using the base field sequence number specified and the increment specified.

If resequencing the fields would result in a number greater than 999, message 152 SEQUENCE NUMBERS WILL NOT FIT is displayed. Repeat the command with a smaller increment.

SAVE

The SAVE command transfers the contents of the workarea to the Formats file.

/SAVE

Use this command to update the Formats file after inserting a new format or making changes to an existing format.

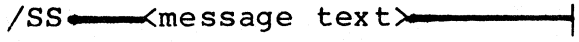
The contents of the workarea remain intact after using the SAVE command, so further changes can be made to the format without using the GET command again. Once the workarea has been saved, you may retrieve other formats from the Formats file without saving or deleting the workarea.

If the Formats file is in use by EXTRACT, message 153, FORMATS FILE IS IN USE, is displayed. Formats cannot be saved while extraction is in progress, but they can be temporarily dumped and then loaded to the Formats file at a later time. Refer to the DUMP and LOAD commands.

The system option "On-Line Format Maintenance" controls whether saving is permitted while EDIT is running. If that option is included, formats can be saved while data is being entered, provided there are no open batches which use the group affected. If a batch is open in that group, message 157, GROUP IN USE BY EDIT, is displayed. If the On-line Format Maintenance option is excluded, message 153, FORMATS FILE IN USE is displayed.

SEND TO SPO (SS)

The SEND TO SPO command sends a message to the ODT.



The message text is truncated if it exceeds 60 characters. Note that a blank character is required between the command and the message text.

This command is helpful when stations are located at some distance from the ODT.

APPENDIX B

SYSTEM PARAMETERS AND FORMAT MAINTENANCE INPUT FORMATS

The first part of this appendix contains the formats used for input through the File Maintenance Program. The second part of this appendix contains the formats used for input through the Format Maintenance Program. The input formats shown here are intended to represent the screen layouts, which actually appear in the upper left corner of the entire screen.

All input is mandatory unless otherwise noted; you must include a value for every parameter. Default values for many options are included with the release, and are shown here within forms delimiters for each option and are stated in the explanatory text following each screen. A blank field signifies that there is no default value and you must enter a value for that parameter.

FILEMAINT INPUT FORMATS

The input formats included here are used to enter system parameters for the File Maintenance Program. Consult Section 4 for more information.

SYSTEM CONFIGURATION PARAMETERS

System configuration parameters consist of parameters referenced by the Edit, Format Maintenance, and Extraction Programs, as well as global parameters affecting the operation of the system as a whole.

EDIT Configuration Parameters

Use this format to specify parameters used by the Edit Program and also to specify several general-purpose parameters. Some of these parameters control the sizes of certain tables held in memory by the Edit Program and therefore affect memory requirements. Keeping the sizes of these tables small increases format overlays, which may affect response time. You may prefer to increase the default values.

```
  000 0000 CONF          [ ] [CON2]
MAINTENANCE TYPE             [CONI]
MAX FORMATS/FIELDS     [06] [0100]
MAX CHARS OF CONST DATA      [0200]
MAX U.P. [00] SHUTDOWN C,T,U [C]
MAX IN SZ/%TNK FULL [0160]   [95]
REC SZ OF TANK FILE:A,B,C    [A]
PAGE SZ/DATE TYP:U,I,E [60]  [U]
```

Maintenance Type

This option determines the type of file maintenance. With the default value CONI, the System file is updated to include the configuration data entered. With the optional value INIT, the Formats, System and Tank files are created and initialized using the configuration parameters entered.

Maximum Formats/Fields

The first value determines the size of the format table for the Edit Program. Enter a 2-digit number from 1 through 63. The default is 6.

The second value determines the size of the field table for the Edit Program. Enter a 4-digit number from 1 through 2047. The default is 100.

Maximum Characters of Constant Data

This is the size of the constant table for the Edit Program. It holds inserted constants, range check constants, and value check constants. Enter a 4-digit number from 100 through 4095. The default is 200.

Maximum User Programs

This 2-digit number determines the size of the user program table for the Edit Program. Enter a number from 0 through 31. The default is 0.

Shutdown Option

This parameter determines when the Data Entry Program sends an end-of-file condition to the user program. Enter "C" if the program is to be stopped when the CLOSE command is received from the last station using it. Enter "T" if the program is to be stopped only when the Data Entry Program terminates. Enter "U" if the user program is to handle the advisory message (message type 98). The default value is "C".

Maximum Input Size/Percentage for Tank File Warning

The first value is the maximum input data size for any format used by the Edit Program. It also affects the Tank file record size, which must be at least 20 bytes larger. Enter a 4-digit number from 1 through 1023. The default is 160.

A warning is given by the Edit Program when the Tank file is some specified percent full and this parameter specifies that percentage. Enter a 2-digit number from 25 through 99. The default is 95.

Tank File Record Size

This parameter specifies the record size of the Tank file. Three values are possible. Choose the smallest that is

consistent with maximum input size. Enter A for 180 bytes, B for 540 bytes, and C for 1080 bytes. The default is A.

Page Size/Date Type

The first value is the page size for all reports printed by all programs. Enter a 2-digit number from 10 through 99. The default is 60 lines per page.

The second value is the format of the date printed in all reports by all programs. It is also the system default date type for checks on date fields. Enter U for USA format (MMDDYY), E for European format (DDMMYY), or I for International format (YYMMDD). The default is U.

FORMAINT and EXTRACT Configuration Parameters

This format consists of parameters for the Format Maintenance and Extraction Programs. These parameters control the sizes of certain tables held in memory by the programs and therefore affect memory requirements. You may prefer to increase the default values.

```
  000 0000 CON2        [ ] [CN3R]
MAX FIELDS - FORMAINT      [063]
MAX CHARS OF CONST DATA    [0500]
MAX CHARS SCRATCHPAD CNS   [0100]
MAX FORMATS - EXTRACT       [10]
MAX FIELDS                 [0200]
MAX CHARS OF CONST DATA    [1000]
MAX OUTPUT RECORD SIZE      [400]
```

Maximum Fields

This 3-digit value determines the size of the field table for the Format Maintenance Program. Since FORMAINT is used to insert format data, this value is also the maximum number of fields in any format. Enter a number from 63 through 255. The default is 63.

Maximum Characters of Constant Data

This 4-digit number is the size of the constant table for the Format Maintenance Program. It holds inserted constants, range check constants, and value check constants. Since FORMAINT is used to insert format data, this value is also the maximum number of characters of constant data in any format. Enter a number from 100 through 4095. The default is 500.

Maximum Characters of Scratchpad Constant Data

This is the size of the scratchpad constant table for the Format Maintenance Program. Since FORMAINT is used to insert format data, this value is also the maximum number of characters of constant data in any field of any format. Enter a 4-digit number from 25 through 1023. The default is 100.

Maximum Formats

This 2-digit number determines the size of the format table for the Extraction Program. Enter a number from 1 through 31. The default is 10.

Maximum Fields

This value is the size of the field table for the Extraction Program. Enter a 4-digit number from 1 through 1023. The default is 200.

Maximum Characters of Constant Data

This 4-digit number is the size of the constant table for the Extraction Program. It is used for field names and inserted constants. Enter a number from 250 through 2000. The default is 1000.

Maximum Output Record Size

This is the maximum output record size for an application file produced by the Extraction Program. Enter a 3-digit number from 80 through 999. The default is 400.

Duplicate and Restrict Configuration Parameters

These parameters allow data to be duplicated from any format and enable certain commands to be restricted to supervisors. In order for the change to take place, a CN3R screen must always be followed by a CN3I screen.

```
  000 0000 CN3R       [ ][CN3I]
DUPLICATE ANY: Y,N            [N]

RESTRICT OPTIONS: R,N
 ACCEPT      [N]  CONTROL     [N]
 DELETE      [N]  INSERT      [N]
 SCHEDULE    [N]  SCHD.DEL    [N]
 EXT.SCHD    [N]  TERMINATE   [N]
```

Duplicate Any

This option allows data to be duplicated from the previous format regardless of whether the format name has changed. Enter Y (yes) for this feature or N (no) for the default option, which duplicates data from the previous format only if the format name is the same.

Restrict Options

The restrict options allow the Edit Program commands indicated to be restricted to operators with supervisory status. If the schedule option is not restricted, but the schedule delete option is restricted, only the delete options of the SCHEDULE command are restricted. Enter R for restricted or N for nonrestricted. The default is N.

Include Configuration Parameters

The include options allow those features and EDIT commands indicated to be excluded from use. This screen should always be used immediately following the CN3R screen.

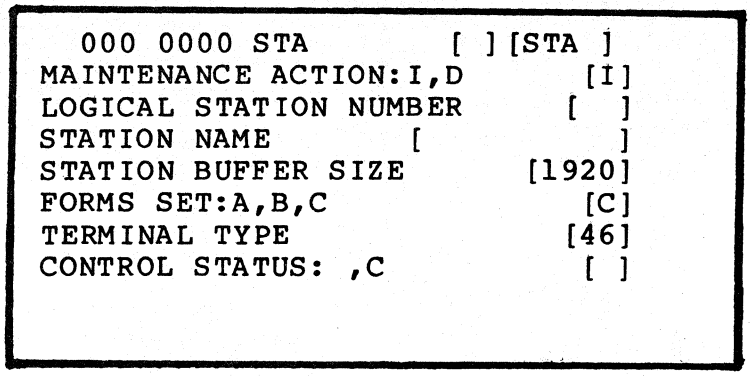
```
  000 0000 CN3I       [ ][STA ]
  ---- OPTIONS: I,E ----
CDV [I] BTOT [I] XFOOT [I]
ON LINE FORMAT MAINTENANCE  [E]
  ---- COMMANDS: I,E ----
ACC [I] CON [I] DEL [I] DIS [I]
INS [I] REA [I] EXT.SCH     [I]
SS  [I] STA [I] WRU [I] DS  [I]
```

If the check digit verification, batch total, or crossfoot options are excluded, the respective checks are not performed by the Edit Program. If the ACCEPT, CONTROL, or SS commands are excluded, they are not recognized by the Edit Program. Enter E for excluded or I for included. The default is I.

If the On-Line Format Maintenance option is included, then saving, loading a specified group, or removing a specified group or format will be allowed while EDIT is running provided the given group is not being used by EDIT. The default for this option is E for excluded.

STATION PARAMETERS

Station parameters are required for each ODESY station in the network.



Maintenance Action

Use this option to specify whether the station is to be inserted or amended, or deleted. Enter I for insert or amend, D for delete. The default is I.

Logical Station Number

Obtain the station number from the NDL listing. Enter a 2-digit number from 1 through 99.

Station Name

Station name is optional. It can be obtained from the NDL listing. Enter a maximum of 10 characters.

Station Buffer Size

Station buffer size is used by ODESY for data communications purposes. Obtain this value from Station Information in Section 2. Enter a 4-digit number from 10 through 4095. The default is 1920.

Forms Set

This option determines the type of forms delimiters used for the station. Enter A for braces ({ }), B for brackets ([ ]), or C for US and RS. The default is C. (ODESY uses US

and RS as forms delimiters on the TD 730, TD 820, TD 830, and MT983.)

Terminal Type

ODESY uses terminal type for data communications purposes. Obtain this value from the station section of the NDL listing. Enter a 2-digit number from 32 through 46. The default is 46. Defined terminal types are:

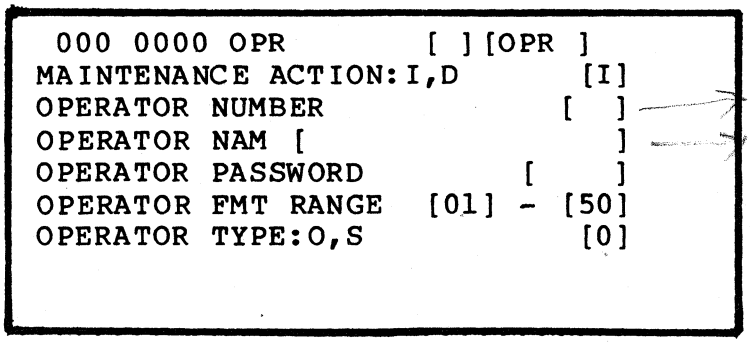
32 = TD 700
33 = TD 730A
34 = TD 730B
41 = TD 801
42 = TD 802
43 = TD 821
44 = TD 822
45 = TD 831
46 = TD 832, MT 983

Control Status

When this option is set the station is a permanent control station, and may use the CONTROL, TERMINATE, and ZIP commands during data entry. Enter C for control status; otherwise leave blank.

OPERATOR PARAMETERS

Operator parameters are required for each ODESY operator.



Maintenance Action

This option determines whether a particular operator is to be inserted or amended, or deleted. Enter I for insert or amend, D for delete. The default is I.

Operator Number

A number is used to identify the operator. Enter a 2-digit number from 1 through 99.

Operator Name

This name is used for documentation purposes only. Enter a maximum of 16 characters for an operator name.

Operator Password

A password is used when logging on for data entry or format maintenance. Enter a maximum of four characters for a password.

Operator Format Range

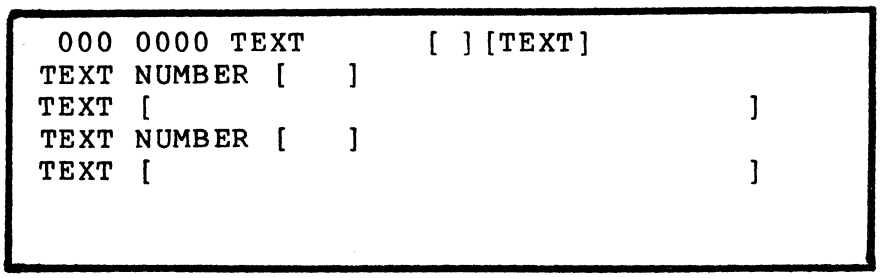
An operator's format range determines what formats the operator can request. The format security level must fall within the operator's range for the format to be displayed. The first value is the lower limit of the range of formats allowed. This is followed by the upper limit. Enter a two digit number from 1 through 99 for each limit, with the lower limit less than or equal to the upper limit.

Operator Type

This value indicates whether the operator has supervisory status. Enter S for a supervisor, 0 for an operator. The default is 0.

MESSAGE TEXT PARAMETERS

The default message texts used in the system messages for the Format Maintenance Program are shown in Appendix D. Use this format to change those default message texts.



Text Number

This is the number used in Appendix D to identify this system message. Enter a 3-digit number from 1 through 199.

Text

Most system messages have a maximum length of 31 characters, but several have fewer characters; refer to Appendix D.

COMMAND PARAMETERS

Only the first three characters of the commands and keywords used for data entry and format maintenance are significant. Default 3-character abbreviations can be changed using these formats. The CMD and CMD2 formats create one ODESYSETUP/INPUT record, and therefore these formats must always be used together. The same is true for the CMD3 and CMD4 formats.

EDIT Command Parameters

Enter a maximum of three characters to identify each command or keyword.

```
  000 0000 CMD          [ ][CMD2]
CLOSE    [CLO] ACCEPT       [ACC]
DELETE   [DEL] INSERT       [INS]
OPEN     [OPE] SCHEDULE     [SCH]
BYE      [BYE] HELLO        [HEL]
STATUS   [STA] SS           [SS ]
WRU      [WRU] TERMINATE    [TER]
CONTROL  [CON] READY        [REA]
```

The DELETE abbreviation is also used as a keyword in the SCHEDULE command.

```
  000 0000 CMD2         [ ][CMD3]
ENTER    [ENT] VERIFY       [VER]
OVERRIDE [OVE] ON           [ON]
OFF      [OFF] NOW          [NOW]
ZIP      [ZIP] ALL          [ALL]
DISPLAY  [DIS] MODIFY       [MOD]
TOTALS   [TOT] INQUIRY      [INQ]
DS       [DS ]
```

The keywords ENTER, VERIFY, and INQUIRY are used with the OPEN command; otherwise these abbreviations appear only in the control line and in status reports to identify the current mode and action. The DELETE command also uses the keywork format, as well as the system text message WORKAREA.

FORMAINT Command Parameters

Enter a maximum of three characters to identify each command or keyword.

```
 000 0000 CMD3        [ ][CMD4]
INSERT    [INS] LIST          [LIS]
AMEND     [AME] COPY          [COP]
MOVE      [MOV] RESEQUENCE    [RES]
DELETE    [DEL] DISPLAY       [DIS]
SAVE      [SAV] GET           [GET]
REMOVE    [REM] PD            [PD]
PRINT     [PRI] HELLO         [HEL]
```

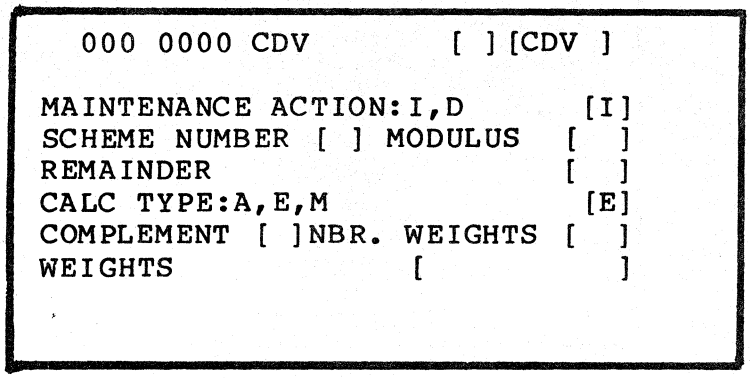
```
  000 0000 CMD4        [ ][     ]
BYE       [BYE] DUMP          [DUM]
LOAD      [LOA] SS            [SS ]

KEYWORDS - GROUP      [GRO]
           FORMAT     [FOR]
           FIELD      [FIE]
```

The keywords GROUP and FORMAT are used with the AMEND, INSERT, and LIST commands, and GROUP, FORMAT, and FIELD appear in the control line to identify the current action. The DELETE command uses the keyword FORMAT and the system text message 116, WORKAREA.

CHECK DIGIT PARAMETERS

You can define a maximum of nine check digit verification schemes. Check digit parameters are required for each check digit verification scheme. Three check digit verification schemes are defined in the standard system configuration as schemes 1, 2, and 3, respectively: Burroughs Modulus 10, IBM Modulus 11, and Burroughs Modulus 11. They cannot be changed directly but can be replaced by deleting the old scheme and inserting a new scheme. Refer to Section 4 and Appendix C for a more detailed explanation of check digit verification.



Maintenance Action

This value determines whether the check digit verification scheme is to be inserted or or deleted. Enter I for insert or D for delete. The default is I.

Scheme Number

This number identifies this check digit verification scheme. Enter a number from 1 through 9.

Modulus and Remainder

The modulus and remainder are used in the calculation of the check digits. For modulus, enter a 2-digit number from 10 through 15. For remainder, enter a 2-digit number from 0 through 15.

Calculation Type

Enter M if the weights are supplied, A if the weights are supplied and all digits are to be added, or E if the products are supplied. The default is E.

Complemented

Complemented determines whether the final product is to be subtracted from the modulus in calculating the check digit.

Enter Y if the check digit is to be complemented; otherwise enter N.

Number of Weights

This is the number of weights used in the check digit calculation. Enter a 2-digit number from 1 through 10 for the number of weights.

Weights

For calculation types M and A, enter the weights as 1-digit hexadecimal numbers; otherwise, enter the products as a group of 10 1-digit hexadecimal products for each weight.

PRINT PARAMETERS

Use this format to request printing of all or part of the parameters in the System file.

```
  000 0000 "B        [ ][PRNT]
PRINT DATA
 [ALL]  [   ]  [   ]
 [   ]  [   ]
```

The first print request is mandatory, but the other four requests are optional.

Enter a valid request type. The default is ALL. Defined print requests are:

ALL = All request types.

CDV = Check digit data.

CMD = Commands data.

CON = Configuration data.

CTL = Control record data from the master files.

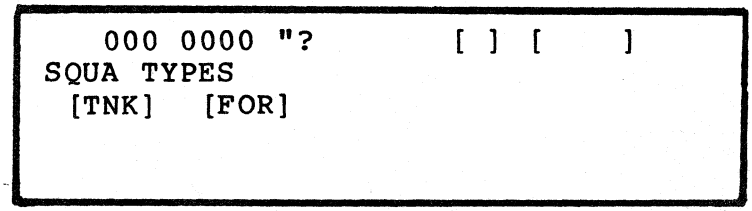
OPR = Operator data.

STA = Station data.

TXT = System message data.

SQUASH PARAMETERS

Use this format to request squashing of the Tank file and/or the formats file.



The first squash request is mandatory, but the second request is optional.

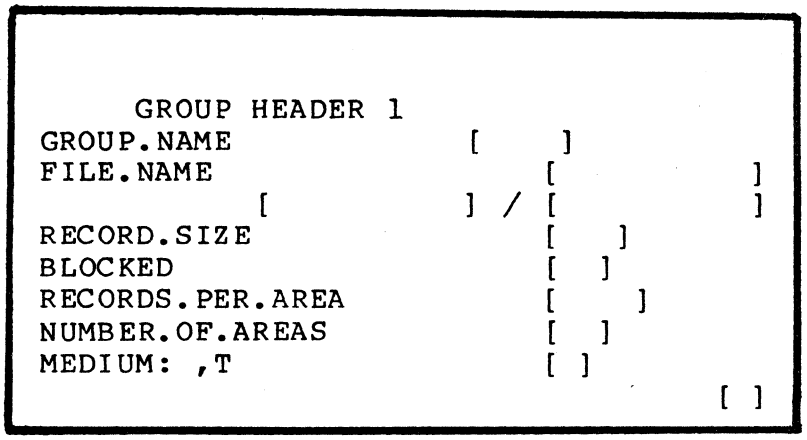
Enter TNK to squash the Tank file or FOR to squash the Formats file. Both values appear as the default.

FORMAINT INPUT FORMATS

The input formats included here are used to create and modify screen formats through the Format Maintenance Program. Consult Section 6 for more information.

GROUP HEADER DATA

Group header data specifies the physical characteristics of the application file and several general options for the group. It links the application file with the formats in the group. Refer to "File Atributes" in the B 1000 Systems System Operation Guide, Volume 1, 1108982.



Group Name

This name identifies the group. Enter a maximum of four alphanumeric (A-Z or 0-9) characters.

File Name

This is the name of the application file produced by ODESY when a batch is created and scheduled using this group. A three-part name is allowed, consisting of pack-identification, family-identification, and file-identification. The file-ID is mandatory only if a usercode is specified in the family name; otherwise only the family name is mandatory. For disk files, the pack-ID is optional. For tape files, the pack-ID must be left blank.

Record Size

This value indicates the record size of the application file. Enter a number from 1 through 999. The default is 180.

Blocked

This value is the blocking factor of the application file. Enter a number from 1 through 99. The default is 1.

Records per Area

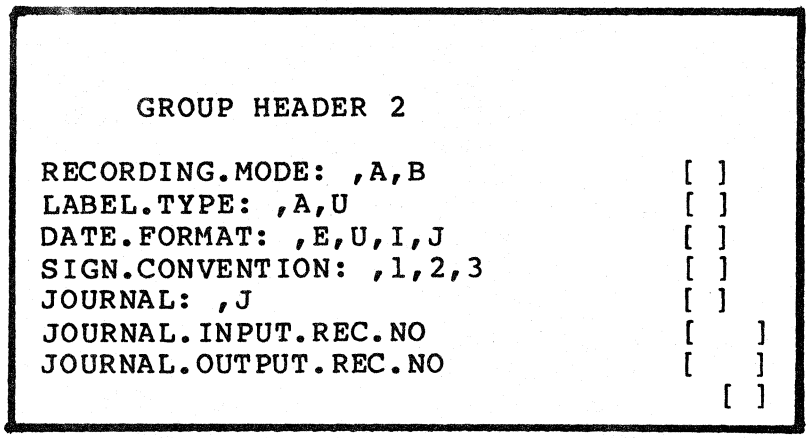
This number is the area size of the application file, but is required only for disk files. Enter a number from 1 through 9999. The default is 500. (This number must be a multiple of the value specified in Blocked.)

Number of Areas

This value is the maximum number of areas allowed for the application file, but is required only for disk files. Enter a number from 1 through 99. The default is 40.

Medium

This option specifies the type of medium on which the application file is stored. Enter T for tape or leave blank for disk. Disk includes any disk. Tape includes all types of tape.



Recording Mode

Set this value only if Tape is specified for Medium. It refers to the representation of the file data as it goes out to tape. Enter A for ASCII, B for BCL, and leave blank for the default of EBCDIC.

Label Type

This value is the label type to be used for the application file, but is required only for tape files. Enter A for ANSI label, U for unlabelled, and leave blank for Burroughs standard label.

Date Format

This option determines the format of the date assumed for all date check fields in formats in this group. Enter U for USA format (MMDDYY), E for European format (DDMMYY), I for International format (YYMMDD), J for Julian format (YYDDD), and leave blank for the system default date type, which is set in the standard system configuration as USA format.

Sign Convention

Sign convention controls the position of the sign in all signed numeric fields which are created using formats in this group. It affects only the application file. Enter 1 for least significant position, 2 for most significant position overpunch, 3 for least significant position overpunch, and leave blank for the default which is most significant position.

Journal

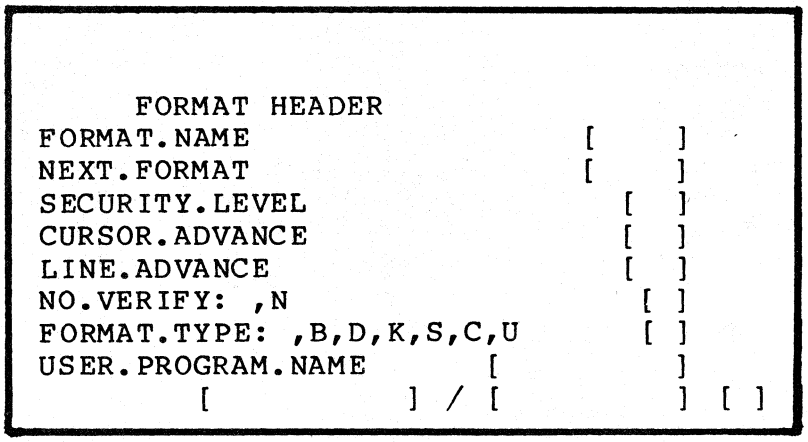
This option must be set if a journal is required for formats in this group. Enter J for journal; otherwise leave blank.

Journal Input Record Numbers and Journal Output Record Numbers

These numbers apply only if the journal option is set. These two options provide for record numbers to be printed on each line of the journal. If there is no journal or record numbers are not to be printed, leave blank; otherwise enter a number from 1 through 127 to specify the print column. Note that each record number requires six digits; ensure that the input and output record numbers are sufficiently spaced so as not to overlap.

FORMAT HEADER DATA

Format header data determines which options are not specific to individual fields of the screen format.



Format Name

This name identifies the format. Enter a maximum of four characters. They can be alphanumeric.

Next Format

This name identifies the logical successor or the format that will normally follow this format. Enter a maximum of four characters, which can be alphanumeric, or leave blank if there is no successor.

Security Level

This value specifies the security level of the format. An operator can access the format only if the security level of the format is within the range of formats specified for the operator. Enter a number from 0 through 99. The default is 50.

Cursor Advance

This option allows the cursor to be advanced automatically from the start of the format during data entry. It specifies the number of fields that will be skipped or tabbed over. Enter a number from 0 through 15. The default is 0.

Line Advance

Line Advance is the number of blank lines left between the control line and and the first field of the format. Enter a number from 0 through 15. The default is 0.

No Verify

This option provides for data which is not to be verified; if the option is set, records are automatically marked as "verified" when they are entered and cannot be verified in Verify mode. Enter N for no verify; otherwise leave blank.

Format Type

This option must be set if this is either a batch header format used for batch total checks or a display-only format. Enter B for a batch header format or D for a display format. The Format Type option is also used to specify format type if this is an inquiry format. Enter K for key-data format, S for supply-data format, C for check-data format, or U for supply-and-check format.

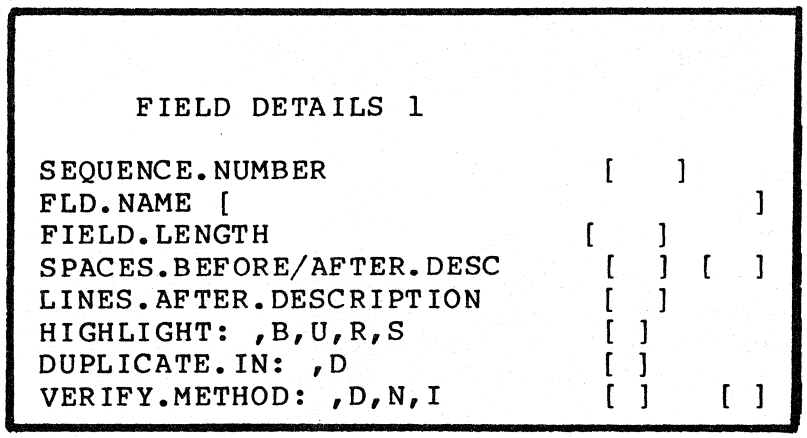
User Program Name

This is the name of the user program that handles this format; it is applicable only to inquiry formats. A pack name, family name, and file name can be specified and, if an inquiry format type has been specified, this option is mandatory. This option must be left blank if the format type is blank.

FIELD DATA

Each field of the format must be specified. Use this first screen format to specify display options.

The Field Screen Layout/Audit and Output Specifications Form, contained at the end of this appendix, presents the same information.



Field Sequence Number

This number, a number from 1 through 999, identifies the field. ODESY provides a default sequence number, beginning at 10 and incremented by 10 for each field.

Field Name

This name is displayed on the screen during data entry and may be printed in the journal. Enter a maximum of 28 characters. It is always truncated by omitting spaces to the right.

Field Length

This number refers to the length of the field. During data entry, a field of this length will be displayed on the screen enclosed in a pair of field delimiters. For a

name-only field leave blank; otherwise enter a number from 1 through 160. The default is 0.

Space Advance Before

Space Advance Before is the number of blank characters before the field name during data entry. Enter a number from 0 through 99. The default is 0.

Space Advance After

This value determines the number of blank characters between the field name and leftmost field delimiter during data entry. For name-only fields, enter a number from 0 through 99; otherwise enter a number from 1 through 99. The default is 1. (Note that when specifying a highlight option of secure video, two spaces will be required after the field name. This is to provide an additional space for a blink character in case of an error during data entry. This prevents secured data from being exposed.)

Line Advance After

This option specifies the number of lines to be advanced after the rightmost field delimiter during data entry. Enter a number from 0 through 15. The default is 0.

Highlight

Highlight applies only to the TD 730 and TD 830. If this option is set, the field will be highlighted. Enter B for bright video, U for underline video, R for reverse video, S for secure video; otherwise leave blank. All highlighting features are available on the TD 830, but only secure video applies to the TD 730.

Duplicate In

This option provides for fields to be duplicated on the screen from record to record. Enter D for duplication; otherwise leave blank.

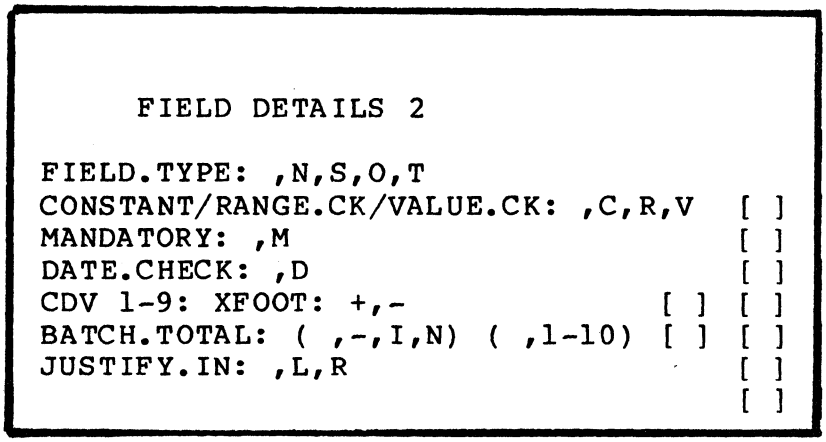
Verify Method

This option allows data within a field to be verified in one of four ways. Leave it blank or enter I for the rekey verification options. These options require that the data be re-entered manually. The difference between these two options involves what is displayed when the field is in error. With the default rekey option, the data keyed most recently is displayed in the error field. With the I option, the data keyed initially, such as during data entry, is displayed.

Specifying a D, for display verify, causes the data to be displayed on the screen during verification. For display verify, the cursor will still tab through the data field. N for no verify causes data to be displayed on the screen during verification and also makes the field transmittable-protected (for terminals that support this function). If you choose this method, the cursor tabs over the field and the field data cannot be changed during verification.

For the first three options, the error fields are blinked (on terminals that support the blink feature). Error fields are simply filled with question marks on terminals that do not support the blink feature.

Use this second screen format to specify audit checks. This screen format will not appear if the field length is zero.



Field Type

This option indicates whether the field is output-only, numeric, or total. If it is output-only, it does not appear on the screen. Output-only fields are always alphanumeric. If it is numeric, a numeric check is performed. A total field is not a data entry field, but rather is designed to display the values of the specified batch total. Enter O for an output-only field, N for an unsigned numeric field, S for a signed numeric field, T for a total field, and leave blank for an alphanumeric field. The maximum field length for an unsigned numeric field is 15. For a signed numeric field or a total field, the maximum field length is 16.

Inserted, Range Check, and Value Check Constants

Set this option if one of the above features is required. Range check constants can be applied only to numeric fields. Enter C for an inserted constant, R for a range check constant, V for a value check constant; otherwise leave blank. The maximum field length for a field with an inserted constant or a value check constant is 28; if it is numeric the lower limit of 15 or 16 applies.

Mandatory Check

Set this option if the field is mandatory, meaning that it may not be blank. Enter M for mandatory check; otherwise leave blank.

Date Check

Date Check applies only to unsigned numeric fields of length 5 for Julian dates or length 6 for other dates and must be set if the field is to be checked as a date. The date type option indicates the order of day, month, and year. Enter D for a date check; otherwise leave blank.

Check Digit Verification

Check digit verification applies only to unsigned numeric fields and must be set if a check digit verification scheme is to be applied to this field. You can define a maximum of nine different check digit verification schemes. Enter a number from 1 through 9 for check digit verification; otherwise leave blank.

Crossfoot Check

These options apply only to numeric fields and must be set if the crossfoot registers are used for subtotalling. There is one crossfoot register and each field can be added to or subtracted from the register. For each option, enter a plus or minus sign if crossfooting is required for the field; otherwise leave blank.

Batch Total Check

There are ten batch total registers that can be used to accumulate totals over many records. Numeric fields can be added to or subtracted from these registers by setting the batch total option to space or "-". The registers can also act as field counters. A specified register can be incremented unconditionally by setting the sign option to "I". If the sign option is set to "N", the register will be incremented only if data has been entered in the field. Enter the batch total number affected in the batch total field. If batch totalling is not required, both fields must be left blank.

Justification Check

Set this option if the field is to be checked for right or left justification. Enter R for right justification, L for left justification; otherwise leave blank.

Use this third screen format to specify output and journal options.

```
     FIELD DETAILS 3

OUTPUT.POSITION                    [   ]
WRITE.AFTER: ,W,C,S                  [ ]
DUPLICATE.OUT: ,D,C                  [ ]
JUSTIFY.OUT: ,L,R; PACKED: ,P,H      [ ]  [ ]
BLANK.FILL: ,B; SIGN: ,+,-           [ ]  [ ]
NAME.COLUMN                        [   ]
DATA.COLUMN                        [   ]
PRINT.AFTER: ,P,S,C                  [ ]  [ ]
```

Output Position

This value indicates the position in the application file where the field is to be placed. If the field is to be output, enter a number from 1 through the record size. If the field is not to be output, leave blank. The output position plus the field length may not be greater than the record size.

Write After

Write After controls the writing of records to the output file. Enter W for write-only, C for write-and-clear, S for save; otherwise leave blank.

Duplicate Out

This option is used in conjunction with the write-after option to duplicate fields in records of the application file. A chained field, unlike a duplicate-out field which may not be altered during data entry until a new format appears, duplicates the value contained within it until a new value is entered in the next chained field of the same format. Enter D for duplication, C for chained field; otherwise leave blank.

Justify Out

Use this option to right-justify or left-justify the field before it is written to the application file. Enter R for right justification, L for left justification; otherwise leave blank.

Packed

This option applies only to numeric fields. If this is a packed numeric (computational) field where each digit is stored as four bits rather than as eight bits, set this option. The output position option is specified in units of eight bits, but this option allows packed numeric fields to begin at either the usual byte boundary or four bits later at a half-byte boundary. Enter P for packed numeric output, H for packed numeric output with half-byte start position; otherwise leave blank.

Blank Fill

This option applies only to numeric fields. With the default, zero fill, blanks are replaced by zeros in numeric fields. Enter B for blank fill; otherwise leave blank.

Sign

This option applies to signed numeric fields. If a field is normally extracted as negative, specify its sign as negative (-). The field will be extracted as negative unless a plus sign (+) is keyed into it during data entry.

Name Column

Name Column applies only if the journal option is set and refers to the position in the journal where the field name is to be printed. Enter a number from 1 through 132 if the field name is to be printed; otherwise leave blank.

Data Column

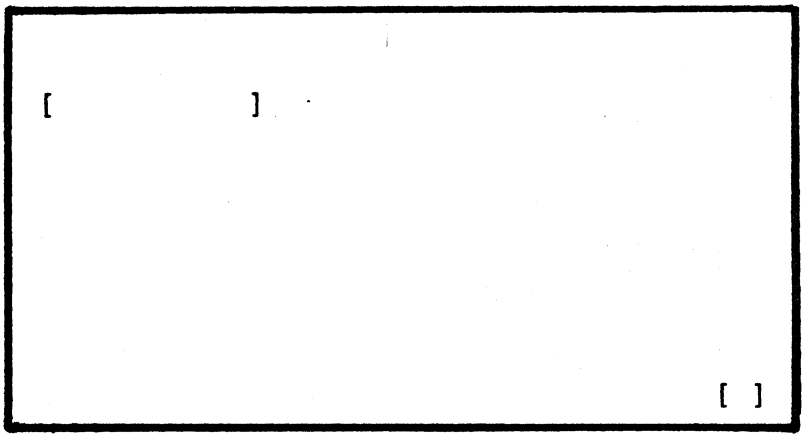
This value applies only if the journal option is set and indicates the position in the journal where the field is to be printed. Enter a number from 1 through 132 if the field is to be printed, or leave blank if it is not to be printed.

Print After

This option applies only if the journal option is set and controls the printing of journal lines. Enter P for print only, S for save, C for print-and-clear; otherwise leave blank.

INSERTED CONSTANT

This screen format appears automatically if the field has an inserted constant.

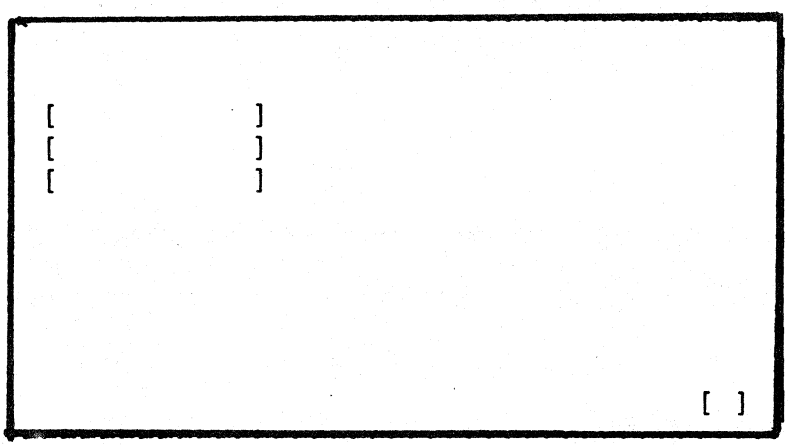


Inserted Constant

Enter a 10-character constant which satisfies the audit checks specified for the field.

RANGE CHECK CONSTANTS

This screen format appears automatically if the field has a range check constant.



First Constant

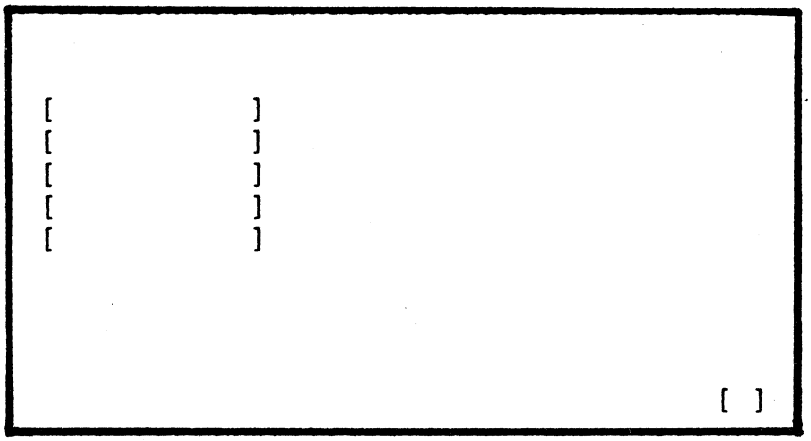
This value is the default for the field. Enter a 10-digit number or leave blank. The number can be signed if the field type is S.

Second and Third Constants

These numbers are the lower and upper limits of the range check. For each constant enter a 10-digit number. The number may be signed if the field type is S.

VALUE CHECK CONSTANTS

This screen format appears automatically if the field has a value check constant.



First Constant

This value indicates the default for the field and is used as the first value check constant. Enter a 10-character or 10-digit constant which satisfies the audit checks specified for the field, or leave blank.

Other Constants

You can enter as many value check constants as the limits of the system (see FORMAINT and EXTRACT configuration data) will allow. If five value check constants are entered, the format is displayed again. The value check list is terminated by a blank constant. Constants must satisfy other audit specifications for the field.

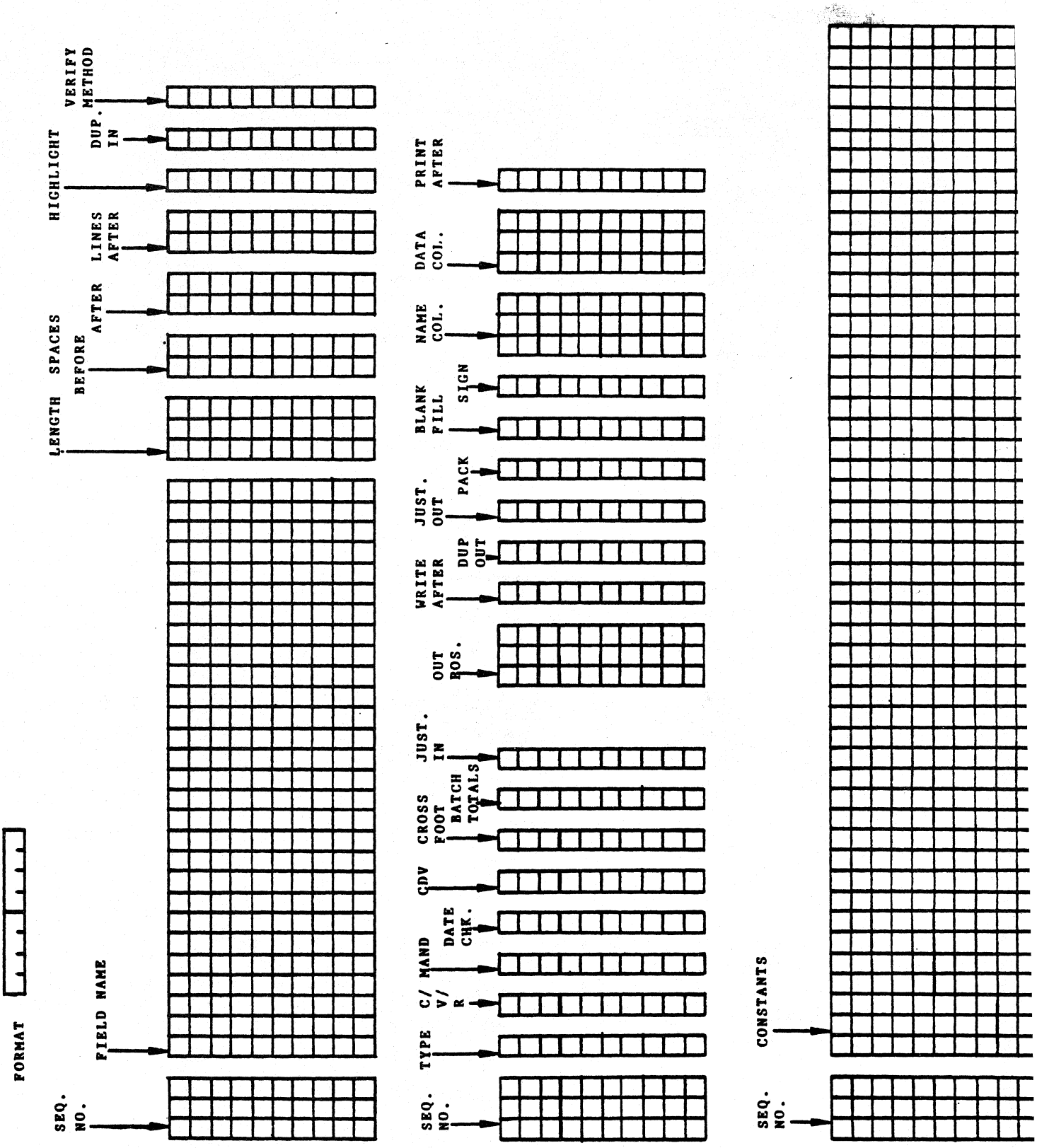


Figure 9. Field Screen Layout/Audit and Output Specifications Form

APPENDIX C

CHECK DIGIT VERIFICATION SCHEMES

Check digit verification is a means of checking or verifying certain numeric data, such as account numbers, through a series of calculations. Check digit verification uses self-checking numbers made up of digits combined in a particular way to produce a fixed result. The self-checking number, known as the check digit, is derived from an ordinary number by adding an extra digit.

For example, convert the account number 12345 to a self-checking number. The check digit calculation is performed on these five digits. In this case, the check digit is 9, so the complete number is 123459. The number is verified by repeating the calculation on the first five digits. For instance, if the operator enters 124359 (interchange of 3 and 4) or 123469 (entry of 6 instead of 5), the check digit is no longer valid and ODESY returns the field as in error.

The most basic check digit scheme is to add the digits together and subtract the total from a multiple of 10. Assuming random errors, this scheme detects nine of 10 mistakes by the operator. However, errors are not completely random. A common error, which this simple scheme does not detect, is to interchange two digits. Most of the common schemes give each digit a different weight, depending on its position in the number. This enables the interchange of digits to be detected.

You cam define a maximum of nine check digit verification schemes, with a choice of:

1. Modulus.
2. Weights (or products).
3. Remainder.
4. Complemented or not complemented.
5. Calculation type (allows individual products to be added).

The most common check digit verification schemes are 1, 2, and 3, as follows, defined in the standard system configuration:

1. Burroughs Modulus 10 and IBM Modulus 10.

Modulus	10
Weights	2, 1, 2, 1, 2, 1, ...
Remainder	0
Calculation Type	"Add"

2. IBM Modulus 11.

Modulus	11
Weights	2, 3, 4, 5, 6, 7, 2, 3, ...
Remainder	0

3. Burroughs Modulus 11.

Modulus	11
Weights	7, 3, 1, 7, 3, 1, ...
Remainder	1

To add schemes or replace the standard schemes, enter check digit scheme parameters. (Refer to Section 4 and Appendix B.) Because of the sensitivity of these check digit schemes, they cannot be modified directly as with other configuration parameters. Rather, the old scheme must be deleted before the new or modified scheme may be inserted.

ODESY uses the following method to evaluate a check digit. This example shows a series of seven steps and may be regarded as a formula for verifying the check digit. The calculations are illustrated according to the Burroughs Modulus 11 scheme. The "add" and "complemented" options, not used in this scheme, are illustrated in principle.

1. Multiply each digit of the number by the corresponding weight to get a set of "products."

   For example, check the number 123459 using the Burroughs Modulus 11 scheme. The basic number is 12345 and the check digit is 9. The Modulus 11 scheme has weights 7, 3, and 1. The first weight is applied to the least significant digit:

Digits	1	2	3	4	5
Weights	3	7	1	3	7
Products	3	14	3	12	35

2. If the calculation type is "add," reduce each product to a single digit by adding the individual digits together.

   The Burroughs Modulus 11 scheme does not use this option; however, as an example, the product 35 would be reduced to 3 + 5 = 8, the product 12 to 1 + 2 = 3, and the product 14 to 1 + 4 = 5. Single digit products remain unaltered.

3. Add the products or reduced products together.

   In the example above the result is:

   3 + 14 + 3 + 12 + 35 = 67

4. Add the "remainder," a fixed constant, to the result of the previous step.

   In the example above the remainder is 1, giving a result of 68.

5. Subtract the result of the previous step from the next highest multiple of the modulus.

   In the example above the next highest multiple of 11 is 77, giving a result of 9.

6. If the check digit is to be complemented, subtract the result of the previous step from the modulus.

   The Burroughs Modulus 11 scheme does not use this option, so the result, 9, is unchanged. If it did use the complemented option, the result would be 2.

7. The check digit is one of the following:

   1. If the result of the previous step is the same as the modulus, the check digit is 0.

   2. If the result of the previous step is greater than 10, the check digit is the result minus 10.

   3. The check digit is the result of the previous step.

In the example above the check digit is 9, which agrees with the value entered.

ODESY also offers the option of entering products rather than weights. This option allows complex schemes to be defined. ODESY employs products in its check digit verification tables because they are easier to use. If you enter products, they are placed directly in the tables, but if you enter weights they must be converted into products.

ODESY uses a table access method and the tables contain modified products. The table for the Burroughs Modulus 11 scheme contains three entries with 10 products in each entry, as shown:

	P0	P1	P2	P3	P4	P5	P6	P7	P8	P9
First = 7	0	7	3	A	6	2	9	5	1	8
Second = 3	0	3	6	9	1	4	7	A	2	5
Third = 1	0	1	2	3	4	5	6	7	8	9

The products are the result of multiplying the weight by 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9, and then reducing each result to a value less than the modulus.

To do this, subtract the largest possible multiple of the modulus from the number. For the Burroughs mod 11 scheme above, 35 (the digit 5 multiplied by the weight 7) is entered into the table as 2, which is 35 reduced by subtracting 33 (11 x 3). After the products are reduced in this manner to less than the modulus, values greater than 9 should be entered as hex digits A through F, representing the decimal digits 10 through 15.

In understanding how ODESY evaluates a check digit, consider the number 12345. ODESY takes P5 from the first entry (2), P4 from the second entry (1), P3 from the third entry (3), P2 from the first entry (3), and P1 from the second entry (3). These are added together totaling 12. The remainder of 1 is added totaling 13. This is subtracted from 22 (twice the modulus) producing a check digit of 9.

If you add products to the check digit verification tables, the calculation type must be set to "enter products," and there must be 10 products for each weight. Therefore, instead of entering the weights 731, you must enter the products 073A629518, and so on.

The following screens illustrate how the Burroughs modulus 11 scheme would be specified if the weights were entered:

```
  000 0000 CDV          [ ] [CDV ]

MAINTENANCE ACTION:I,D          [ I]
SCHEME NUMBER [4]  MODULUS      [11]
REMAINDER                       [ 1]
CALC TYPE:A,E,M                  [E]
COMPLEMENT  [N]NBR. WEIGHTS     [ 3]
WEIGHTS                  [073A629518]
```

```
  000 0000 CDV          [ ] [CDV ]

MAINTENENCE ACTION:I,D           [I]
SCHEME NUMBER [4] MODULUS       [  ]
REMAINDER                       [  ]
CALC TYPE:A,E,M                  [ ]
COMPLEMENT [ ]NBR. WEIGHTS      [  ]
WEIGHTS                  [0369147A25]
```

```
  000 0000 CDV          [ ] [CDV ]

MAINTENANCE ACTION:I,D           [I]
SCHEME NUMBER [4] MODULUS       [  ]
REMAINDER                       [  ]
CALC TYPE:A,E,M                  [ ]
COMPLEMENT [ ]NBR. WEIGHTS      [  ]
WEIGHTS                  [0123456789]
```

APPENDIX D

FORMAT MAINTENANCE PROGRAM MESSAGES

This appendix contains a list of error and information messages pertaining to the Format Maintenance Program. An explanation is included with the number and standard text of each message. Some messages, listed here individually, comprise the first or second part of the final message which actually appears on the screen.

The text of these messages can be modified through the system parameters input formats used with the File Maintenance Program. The length number, included at the end of each explanation, is the maximum number of characters which can be altered.

NUMBER	MESSAGE	EXPLANATION
101	GROUP	This appears in the first line of control information for the Group Templates. Length 7.
102	FORMAT	This appears in the first line of control information for the Format Template. Length 7.
103	FIELD	This appears in the first line of control information for the Field Templates. Length 7.
104	I-CONS	This appears in the first line of control information for the Inserted Constant Template. Length 7.
105	R-CONS	This appears in the first line of control information for the Range Constants Template. Length 7.
106	V-CONS	This appears in the first line of control information for the Value Constants Template. Length 7.

107	INSERT	This appears in the first line of control information during insertion of format data. Length 7.
108	LIST	This appears in the first line of control information during listing of format data. Length 7.
109	AMEND	This appears in the first line of control information during amendment of format data. Length 7.
110	DISPLAY	This appears in the first line of control information during format display. Length 7.
111	REMOVE	This is the first part of the reply to a REMOVE command. The second part is message 130. Length 7.
112	DUMP	This is the first part of the reply to a DUMP command. The second part is message 130. Length 7.
113	LOAD	This is the first part of the reply to a LOAD command. The second part is message 130. Length 7.
114	PRINT	This is the first part of the reply to a PRINT command. The second part is message 130. Length 7.
116	WORKAREA	This is the first part of the reply to a PRINT WORKAREA command. The second part is message 127. Length 8.
117	ALL	This is the first part of the reply to a DUMP, LOAD, or PRINT command without group or format specifications. The second part is either message 125, 126, or 127. Length 8.

118	INSERTED	This is the second part of the reply to insertion of a field. The first part is message 103. Length 15.
119	AMENDED	This is the second part of the reply to amendment of a group, format, or field. The first part is either message 101, 102, or 103. Length 15.
120	DELETED	This is the second part of the reply to deletion of a group, format, or field. The first part is either message 101, 102, or 103. Length 15.
121	NOT PRESENT	This is the second part of the reply to an AMEND, COPY, DELETE, LIST, or MOVE command when the group, format, or field is not in the workarea. The first part is either message 101, 102, or 103. Length 15.
122	DUPLICATE FIELD	This is the second part of the reply to an INSERT <field> command when the field sequence number entered already exists in the workarea. The first part is message 103. Length 15.
123	SAVED	This is the second part of the reply to a SAVE command. The first part is the group name and format name. Length 15.
124	REMOVED	This is the second part of the reply to a REMOVE command. The first part is the group name and format name. Length 15.
125	DUMPED	This is the second part of the reply to a DUMP command. The first part is the group name and format name. Length 15.

126	LOADED	This is the second part of the reply to a LOAD command. The first part is the group name and format name. Length 15.
127	PRINTED	This is the second part of the reply to a PRINT command. The first part is the group name and format name. Length 15.
128	NOW AVAILABLE	This is the second part of the reply to a GET command. The first part is the group name and format name. Length 15.
129	NOT IN FILE	This is the second part of the reply to a COPY, DUMP, GET, PRINT, or REMOVE command when the group or format names cannot be found in the Formats file. The first part is the group name and format name. Length 15.
130	STARTED	This is the second part of the initial response to a DUMP, LOAD, PRINT, or REMOVE command. The first part is either message 111, 112, 113, or 114. Length 15.
131	DONE	A COPY, MOVE, or RESEQUENCE command has been accomplished successfully. Length 15.
132	NOT RECOGNIZED	The last command was entered incorrectly. Length 30.
133	NOT BUSY	The last input was data rather than a command, but the operator is not inserting, listing, or amending. Length 30.
134	SAVE OR DELETE WORKAREA	Save or delete the workarea before entering a BYE or INSERT command. Length 30.
135	MESSAGE ZIPPED TO MCP	This is the reply to any input preceded by a question mark. Length 30.

136	START OF SESSION	This is the reply to a HELLO command with a valid operator number and password. Length 30.
137	END OF SESSION	This is the reply to a BYE or command. Length 30.
138	ENTER OPERATOR/PASSWORD	No operator is logged on. Length 30.
139	INVALID OPERATOR/ PASSWORD	This is the reply to a HELLO command with an invalid operator number or password. Length 30.
140	ALREADY LOGGED IN	This is the reply to a HELLO command when an operator is already logged on. Length 30.
141	TOO MANY FIELDS FOR FORMAT	This is the reply to an INSERT <field> command when the maximum number of workarea fields has been reached (normally 63 fields). Length 30.
142	GROUP INDEX IS FULL	This is the reply to a LOAD or SAVE command when there are 196 groups in the Formats file. Length 30.
143	CONSTANT TABLE OVERFLOW	This is the reply to the entry of format data when there are too many constants for the workarea. Length 30.
144	NO FORMATS HAVE BEEN LOADED	Either the format or formats specified in the LOAD command were not found in the Dump file, there was a version mismatch (i.e. the formats were dumped by a previous, incompatible version of FORMAINT), or some other difficulty prevented the load. Length 30.
145	FILE IS INCOMPATIBLE	The Dump file specified in the DUMP or LOAD command does not contain records of length 60 bytes, blocked 3. Length 30.

146	FILE IS ALREADY IN USE	The Dump file specified in the DUMP or LOAD command is in use by another program. Length 30.
147	FILE NOT ON DISK	The Dump file specified in the LOAD command is not on disk. Length 30.
148	MESSAGE SENT TO CONSOLE	The message specified in the SS command has been sent to the ODT. Length 30.
149	NOT AN ODESY STATION	The operator has attempted to log on at a station not defined in the ODESY System file. Length 30.
150	OPERATOR LOGGED IN ELSEWHERE	Only one operator may be logged on to FORMAINT at a time. Length 30.
151	SEQUENCE NUMBER ERROR	Sequence numbers in a COPY, MOVE, or RESEQUENCE command have been entered incorrectly. Length 30.
152	SEQUENCE NUMBERS WILL NOT FIT	A COPY, MOVE, or RESEQUENCE command has been entered, but the renumbered fields will not fit in the new location. Length 30.
153	FORMATS FILE IS IN USE	A LOAD, REMOVE, or SAVE command has been entered, but the Formats file is in use by another program. To use certain forms of these commands even while EDIT is running, see the include configuration parameter On-line Format Maintenance in Appendix B, and the command information in Appendix A. Length 30.
154	WARNING, 0 VALUE IN WRITE.AFTER	A SAVE command has been entered and the write-after option for the last field in the format was left blank. The format is saved regardless of the warning. Length 30.

155	FORMAT TOO LARGE TO DISPLAY	A DISPLAY command has been entered, but the format is too large to fit on the screen. Length 30.
156	PLEASE RESTART FORMAINT	The terminal operator should restart FORMAINT the same way it was started originally. This occcurs when FORMAINT has to reconfigure itself and stop. The second time it is executed, it will have the correct amount of memory and will continue to run. Length 23.
157	GROUP IN USE BY EDIT	A LOAD, REMOVE, or SAVE command has been entered, but the group affected is in use by EDIT.

APPENDIX E

EXTRACTION PROGRAM SYSTEM ERRORS

Message Printed	Message Number	Explanatory Notes
BATCH DELETED	955	The deletion of this batch, which was interrupted in a previous extraction run, has now been completed. (Part of recovery.)
BATCH EXTRACTED	958	If the previous extract run aborted while processing this batch, leaving it marked extracted and not marked in-use, this message is logged.
BATCH NOT EXTRACTED	960	The entire batch could not be extracted due to a problem with one or more records in the batch. In this case, an already-existing disk file was being extended, so the records that had been added were removed from the output file. Please correct the problem and re-extract the batch. (To restore the output file to its original size, EXTRACT copies it. Therefore, the MCP message "<output file name> removed" may appear on the ODT when the original output file is removed and the smaller copy is closed.)
EOF ON FORMATS FILE	904	A bad key supplied for a read on the file caused an attempted read beyond the end of the file. See "* NOTE *" at end.

EOF ON OPSTATS FILE	921	See 'EOF ON FORMATS FILE' explanation.
EOF ON SYSTEM FILE	908	See 'EOF ON FORMATS FILE' explanation.
EOF ON TANK FILE	909	See 'EOF ON FORMATS FILE' explanation.
EOF ON TANK(BIX)	911	EOF on tank file encountered while attempting to read a batch index record. See "* NOTE *" at end.
F <FORMAT NAME> MISSING	953	The format through which the current record is being extracted is either not on the formats file, is not readable from the file, or its INPUT.DATA.SIZE exceeds the tank file record size - 20.
FATE UNKNOWN	956	This batch is marked as needing recovery, but is not flagged as in use or scheduled for extraction.
GROUP MISSING	950	The group through which the current batch must be extracted is not on the formats file, or cannot be read successfully from the formats file.
INVALID OUTPUT POSITION	900 901	A field's output position plus its length exceeds the output record size. Please adjust the field specification and rerun EXTRACT.
INVALID PRINT POSITION	902 903	A field's "data column" position on the journal plus field length, or "name column" position plus field name

		length, exceeds the width of the journal line (132 characters). Please adjust the field specification and re-run EXTRACT.
INV READ ON OPSTATS FILE	920	When on-line with EDIT, EXTRACT's request for updated statistics is ignored or denied. (Occurs after completion of scheduled extractions or deletions). See "* NOTE *" at end.
INV REC TYPE-TANK	910	An Invalid record type was read on the tank file. See "* NOTE *" at end.
INV REC TYPE-TANK(BIX)	912	An invalid record type was encountered when a "BIX" record type was expected on reading the tank file. See "* NOTE *" at end.
INV REC TYPE-TANK(DAT)	913	An invalid record type was encountered when a "DAT" record was expected on the tank file during the extraction (as opposed to deletion) of a batch. See "* NOTE *" at end.
INV REC TYPE-TANK(DELETE)	914	An unexpected record type was encountered when reading the tank file during a delete. See "* NOTE *" at end.
MAX OUTPUT SIZE EXCEEDED	962	The record being extracted exceeds the maximum output record size specified in the system configuration parameters.

NO SPACE FOR FLDS OR CNSTS	906 961	The number of fields or constants for the format being loaded exceeds the system configuration parameter maximum fields or maximum constants for EXTRACT. Please adjust the system configuration parameter being exceeded and re-run EXTRACT.
NOT EXTRACTED	957	During recovery, if the batch marked for recovery is not flagged in-use and is still scheduled for extraction, this message is logged. EXTRACT then proceeds as with a normal extraction run, beginning with the first batch. This batch should be extracted in turn.
OUTPUT FILE FULL	959	The end-of-file for the output file was encountered, preventing the extraction of any more records to that file.
OUTPUT FILE IN USE	951	The output file to which the batch will be extracted already exists and is in use by another program.
OUTPUT FILE INCOMPATIBLE	952	The output file to which the batch will be extracted is already out on disk (under the name specified in the group header), but has a different record size or blocking factor than the file specified in the group header.
OUTPUT FILE PURGED	954	The output file, newly-created for the extraction of the current batch, was closed with purge (removed from disk) because the current batch's extraction could not be completed.

UNEXP REC TYPE-FORMATS F.	905	An unexpected record type was encountered on a read of the formats file. See "* NOTE *" at end.

* NOTE *

These system errors will produce a memory dump. Consider copying current data files (ODESYDATA/=) to storage media and submitting them along with an FTR. For a complete recovery, it may be necessary to reload backup data files.

APPENDIX F

INTERNAL AND EXTERNAL FILE NAMES

The following is a list of internal and external file names by program:

INTERNAL NAME	EXTERNAL NAME	TYPE
ODESY/FILEMAINT		
SYSTEM	ODESYDATA/SYSTEM	Disk
TANK	ODESYDATA/TANK	Disk
FORMATS	ODESYDATA/FORMATS	Disk
LINE	LINE	Printer
COPY	COPY	Disk
FORCOPY	FORCOPY	Disk
WORK	ODESYWORK / #	Disk
INPUT	INPUT	Disk
ODESY/FORMAINT		
LOG	LOG	Printer
LINE	LINE	Printer
FORMATS	ODESYDATA/FORMATS	Disk
SYSTEM	ODESYDATA/SYSTEM	Disk
DUMPFILE	DUMPFILE	Disk
RMTE	STATIONS	Remote
FORMAINTIN	FORMAINTIN	Queue*
FORMAINTOUT	FORMAINTOUT	Queue*

ODESY/EDIT

SYSTEM	ODESYDATA/SYSTEM	Disk
TANK	ODESYDATA/TANK	Disk
FORMATS	ODESYDATA/FORMATS	Disk
STATIONS	STATIONS	Remote
LINE	LINE	Printer
OPSTATS	ODESYDATA/OPSTATS	Disk
EDITIN	EDITIN	Queue*
EDITOUT	EDITOUT	Queue*
FORMAINTIN	FORMAINTIN	Queue*
FORMAINTOUT	FORMAINTOUT	Queue*

ODESY/EXTRACT

LINE	LINE	Printer
LOG	LOG	Printer
FORMATS	ODESYDATA/FORMATS	Disk
SYSTEM	ODESYDATA/SYSTEM	Disk
TANK	ODESYDATA/TANK	Disk
OPUT	OUTPUT	Disk
COPY	OUTPUT	Disk
OPSTATS	ODESYDATA/OPSTATS	Disk
EDITIN	EDITIN	Queue*
EDITOUT	EDITOUT	Queue*

ODESY/STATS

LINE	LINE	Printer
FORMATS	ODESYDATA/FORMATS	Disk

SYSTEM	ODESYDATA/SYSTEM	Disk
ODESY/PRINT		
LINE	LINE	Printer
TANK	ODESYDATA/TANK	Disk
SYSTEM	ODESYDATA/SYSTEM	Disk
ODESY/DUMP		
LINE	LINE	Printer
LOG	LOG	Printer
FORMATS	ODESYDATA/FORMATS	Disk
SYSTEM	ODESYDATA/SYSTEM	Disk
TANK	ODESYDATA/TANK	Disk
ODESY/CONVERT		
LINE	LINE	Printer
DUMP	DUMP	Disk
SYSTEM	ODESYDATA/SYSTEM	Disk
ODESY/TANKCONV		
SYSTEM	ODESYDATA/SYSTEM	Disk
TANK	ODESYDATA/TANK	Disk
FORMATS	ODESYDATA/FORMATS	Disk
LINE	LINE	Printer
COPY	COPY	Disk
COPY.SYS	COPY.SYS	Disk

* If more than one copy of ODESY is being run, queue file names as well as data file names should be modified to be unique for each set of programs.

APPPENDIX G

MEMORY REQUIREMENTS FOR ODESY PROGRAMS

Several system configuration parameters are used to determine the memory requirements for ODESY programs. The parameters and their impact on memory requirements are specified in the dynamic memory formulas below. For additional information on each of the system configuation parameters used, see Section 4 and Appendix B.

EDIT

The minimum memory requirements are calculated by adding 17018 bytes to dynamic memory. For EDIT, dynamic memory is calculated as follows:

```
  <Tank record size> * 2
+ <max terminal buffer size> + 70
+ <max input size> * (<stations> + 1)
+ <stations> * 114.375
+ <max formats for EDIT> * 22.87
+ <max fields for EDIT> * 39.875
+ <characters constants for EDIT>
+ (<user programs> + 1) * 33.125
```

As an example, assume that there are four TD 830 terminals (1920-character screen), six formats, 200 fields, 500 characters constant data, no user programs, a maximum input of 160, and a Tank record size of 180. The minimum amount of memory needed by EDIT is calculated as follows:

```
  17018
+ 180 * 2
+ 1920 + 70
+ 160 * 5
+ 4 * 114.375
+ 6 * 22.87
+ 100 * 39.875
+ 500
+ 1 * 33.125

=25284 bytes
```

FORMAINT

The minimum memory requirements are calculated by adding 19092 bytes to dynamic memory. For FORMAINT, dynamic memory is calculated as follows:

```
  <max terminal buffer size> + 50
+ <max fields for FORMAINT> * 46.375
+ <characters constant data for FORMAINT>
+ <characters scratchpad constant data for FORMAINT>
```

For example, assume that there is one TD830 (1920-character screen) with maximum values of 63 fields per format, 500 characters constant data per format, and 100 characters constant data per field. The minimum memory needed by FORMAINT is calculated as follows:

```
  19092
+ 1920 + 50
+ 63 * 46.375
+ 500
+ 100

= 24584 bytes
```

EXTRACT

Minimum memory requirements are calculated by adding 18206 bytes to dynamic memory. For EXTRACT, the dynamic memory is calculated as follows:

```
  <Tank record size>
+ <max formats for EXTRACT> * 13
+ <max fields for EXTRACT> * 38.625
+ <max constants for EXTRACT>
+ <max output size> * 2
```

For example, assume that there is a Tank record size of 180, with 10 formats, 200 fields, 1000 contents, and an 80-byte output buffer. The minimum memory needed by EXTRACT is calculated as follows:

```
  18206
+ 180
+ 10 * 13
+ 200 * 38.625
+ 1000
+ 80 * 2

=27401 bytes
```

INDEX

Revised 2-17-83 by
PCN 1131851-001