UNISYS

MAPPER® System Manual Functions

Operations Reference Manual

May 1990

Priced Item

Printed in U S America
7831 9266–000

UNISYS

MAPPER® System Manual Functions

Operations Reference Manual

May 1990

Printed in U S America

Priced Item
7831 9266–000

The names, places, and/or events used in this publication 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.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THE DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special, or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this publication should be forwarded using the Business Reply Mail form in this document, or remarks may be addressed directly to Unisys Corporation, MAPPER Product Information, P.O. Box 64942 MS: 4792, St. Paul, Minnesota, 55164-0942, U.S.A.

Page Status

Page	Issue
iii through xv	Original
Contents tab	Original
xvii through xxvii	Original
xxix	Original
xxxi	Original
Section 1 tab	Original
1-1 through 1-3	Original
Section 2 tab	Original
2-1 through 2-29	Original
Section 3 tab	Original
3-1 through 3-21	Original
Section 4 tab	Original
4-1 through 4-8	Original
A - C tab	Original
4-9 through 4-124	Original
D - O tab	Original
4-125 through 4-194	Original
P - V tab	Original
4-195 through 4-270	Original
Section 5 tab	Original
5-1 through 5-6	Original
Appendixes tab	Original
A-1 through A-23	Original
B-1 through B-15	Original
C-1 through C-9	Original
D-1 through D-4	Original
E-1 through E-4	Original
F-1 through F-13	Original
G-1 through G-16	Original

Page	Issue
Glossary tab	Original
Glossary-1 through Glossary-38	Original
Index tab	Original
Index-1 through Index-29	Original

About This Manual

Purpose

This manual provides reference information for using MAPPER system manual functions. It includes an overview of the database structure, comprehensive descriptions and examples of some of the MAPPER system functions. Descriptions and examples for those functions not covered in this manual can be found in the online help system (HELP,*call*).

Scope

This manual presents MAPPER manual functions alphabetically and provides examples based on the demonstration database and on sample databases.

Audience

This manual is for all users of MAPPER manual functions.

Prerequisites

Before using this manual, you should also have a basic understanding of MAPPER terms and concepts. You should be comfortable with using MAPPER software and be familiar with the demonstration database. It is assumed that you have read the *Manual Functions Training Guide* and have completed the tutorial exercises in that guide.

How to Use This Manual

To aid your understanding of this manual and your use of MAPPER software, certain style conventions are used. Following is a description of how this manual handles special characters, system-specific information, examples, keyboard key names, uppercase letters, italics, and color.

After reading this section, please refer to Appendix C for a description of the key names used in this manual.

Then read Sections 1 and 2, which provide background information for using reports and MAPPER manual functions.

Section 3 provides an overview of MAPPER functions and runs grouped by the type of operation performed.

Section 4 is the main reference for MAPPER manual functions. It is alphabetized by function call. Refer to Section 4 for comprehensive information on most functions, as well as examples. Information for other functions can be found in the online help system (HELP,*call*).

If you cannot find the section for a particular function or run, check "List of Functions and Runs" in Section 4, which lists the functions and runs alphabetically by name rather than by function call and indicates if it is online only.

Special Characters

The following characters have special meanings in this document:

[ ] Used around optional entries. If you select the entry, do not type the brackets, just the character or characters inside them.

| Represents the tab character in some screens and formats. Otherwise, the tab is displayed as a center dot (·).

▮ Represents the cursor on your screen. This shows where you type information.

♦ Represents the start-of-entry (SOE) character on your screen.

{ } Used to show a selection of required fields.

| Delimits items enclosed in brackets or braces. You choose one of the items to enter, but you do not enter the brackets, braces, or vertical bars.

Δ Indicates a required space in the format of a command. This character is used where the spaces in the format are not shown clearly.

System-Specific Information

Because MAPPER software is available on many different hardware systems, certain portions of this document may contain information that applies only to one system. One library is used for all systems for these reasons:

- Seeing information that differs with each system helps you port applications from one system to another.
- Having one set of manuals rather than a separate set for each system reduces the cost to you.

This system-specific information can be recognized by its green color, a triangle placed next to the text, and a phrase identifying which hardware system the information applies to.

The following example is a piece of text that applies only the BTOS II MAPPER System:

▽ **BTOS:** Only six languages (1 - 6) are supported.

Abbreviations for Different Hardware Systems

The following list includes the abbreviations used next to triangles in the MAPPER library, along with the full names of the hardware systems.

Abbreviation	System Name
A Series	A Series MAPPER System
BTOS	BTOS II MAPPER System
1100	OS 1100 MAPPER System
PC MAPPER	Personal Computer MAPPER System
U Series	U Series MAPPER System
UNIX®	MAPPER Systems that run on UNIX systems not supplied by Unisys

Examples

The examples in this document appear in lowercase bold letters. Unless otherwise specified, you can type them in lowercase or uppercase letters.

Whenever possible, examples are created and tested against the demonstration database that comes with your system. You can try these examples yourself and receive the same results as shown in this document.

Caution

Be careful when trying examples that update the demonstration database. Do not use functions that add, delete, or change lines while working with report 2B0, 1C0, or 1D0. Use duplicate reports so other users start with the same basic information and get the same results as shown in this document.

UNIX is a registered trademark of AT&T Information Systems.

Screen Illustrations

To help you understand concepts and procedures, this guide uses screen illustrations that closely resemble actual screen displays. However, you may encounter some differences between the illustrations and screen displays. These differences are due to variations among MAPPER systems, among terminal types, and among characteristics of the databases of each MAPPER system. These are some of the differences you might find:

- Vertical bars (|) shown between fields in screen illustrations may or may not appear on your screen display, depending on your MAPPER system and terminal settings. Also, the SOE character (♦) may be a different symbol on your screen.
- Function key bars may contain different key names. For example, if you are using an OS 1100 MAPPER System, the SOE key does not appear on the function key bar.
- Key names within screen text may be different. For example, a screen illustration might mention the **Transmit** key; on the actual screen, it is displayed as XMIT, F5, or whichever key is applicable to your terminal.
- Title lines and date lines may be positioned differently and contain different information. For example, on an OS 1100 MAPPER System, the title line might begin with an indicator called a save flag (such as @991231); save flags are not shown on the screen illustrations in this document.
- The report numbers, dates, times, and user-ids shown on screen illustrations may be different from those you actually see on your screen.

Obtaining Online Help for Keys

To obtain a list of some of the most common generic key names used in the documentation, press **KeyHlp** from the first screen that appears when you enter the MAPPER system.

Some MAPPER systems also provide a list of actual key sequences to use on your keyboard for various key actions. If this feature is available, you will see an entry in the first **KeyHlp** screen called **KeyMap**. Press the keys shown to the right of the word **KeyMap** to display this list of actual key sequences.

If **KeyMap** does not appear on the first **KeyHlp** screen, specific information for your keyboard is not available online. Refer to the printed documentation for your terminal.

You can also use the following methods to obtain the list of generic key names:

Method 1

1. Press **Help** until you reach the main help menu.
2. Tab to "Key Assignments for Documentation Key Names."
3. Transmit.

Method 2

1. After signing on to the MAPPER system, move the cursor to the control line.
2. Type `help,key`.
3. Transmit.

Uppercase Letters

These items appear in uppercase letters:

- MAPPER function calls (for example, CHG)
- MAPPER runs (for example, NAME run)
- Reserved words (for example, DATE1$)

Italics

Italics indicate the following values that you supply:

- The italicized letter N (*n* or *N*) stands for a numeral (*nn* stands for two digits, *nnn* for three, and so on).
- Other italicized letters (for example, *x* and *y*) indicate user-supplied variables.

Color

The color blue is used to indicate formats and anything you are instructed to type. For example:

> To start the ICAL run, enter `ical` on the control line.

The color green, combined with a triangle in the margin, indicates that the information is specific to a certain system. The color stops at the end of the system-specific information. For example:

> ▽ **BTOS:** Only six languages (1 - 6) are supported.

Organization

This manual contains five sections and seven appendixes:

Section 1. Introduction

This section defines a manual function.

Section 2. MAPPER Software: An Overview

This section describes MAPPER software and the structure of the database. It tells how to execute MAPPER software and how to maintain security.

Section 3. MAPPER Function Finder

This section provides an overview of MAPPER functions and runs grouped by operation type.

Section 4. MAPPER Functions and Runs

This section presents MAPPER functions and runs alphabetically by the function call.

1100: Section 5. Iterative Runs

This section describes iterative runs and how they are used. It also provides an exercise to help in understanding.

Appendix A. Summaries: Functions and Runs

This appendix summarizes each of the calls, lists the control-line format, and the options available for the most common functions.

Appendix B. Character Sets and Sorting Orders

This appendix lists the ASCII character set, the Fieldata character set, and the order in which characters are sorted with and without the C(S) option.

Appendix C. Using Your Keyboard and Mouse

This appendix defines the names of keys referred to in this manual. It explains where to find a list of specific keys that equal the key names. The appendix also decribes how to use a mouse with MAPPER software.

Appendix D. Control Commands for Transferring Data

This appendix lists the data control commands available that let you control the format of the data in files being transferred.

Appendix E. Mathematical Operators and Functions

This appendix defines mathematical operators and their relative priorities. It also lists mathematical and trigonometric functions.

Appendix F. Application Power Tools

This appendix provides an overview of APT runs that help you design, implement, and maintain your application.

A Series: Appendix G. Using Your Workstation

This appendix explains how to customize your workstation to help you work most efficiently.

Related Product Information

Because MAPPER software contains features for every kind of user, there is an entire library of manuals available. But that does not mean every manual is right for you. Which manuals you need depends on the way you use the software. For example, to perform basic MAPPER functions, you need only the *Manual Functions Training Guide*, but to configure and manage the system for a group of users, you need several manuals.

The following chart shows which manuals fit your needs. For some tasks, several manuals are marked. Before performing these tasks, you need to understand the material in all the manuals marked.

Note: *Some of the documents in this chart may not be included in the MAPPER library for your system. See "For More Information" for instructions on how to obtain a list of documentation in your MAPPER library.*

If you want to ↓ / Then read →	Manual Functions Training Guide	Manual Functions Reference	Run Design Training Guide	Run Design Reference	Color Graphics Guide	Word Processing Guide	Installation Guide	Administration Guide
Read an overview of MAPPER software	●							
Learn to use fundamental MAPPER functions	●							
Design a database and create reports	●							
Know about manual functions and how to use them	●	●						
Learn the fundamentals of run design	●		●					
Know about run statements and how to use them	●	●	●	●				
Design a run-driven application	●	●	●	●				
Learn how to use MAPPER color graphics	●				●			
Learn how to use MAPPER word processing						●		
Install MAPPER software							●	●
Configure and manage the MAPPER system	●	●	●	●	●	●	●	●

For More Information

To obtain a list of the complete title, the document number, the short title, and a brief description of each manual in the MAPPER library and any other related Unisys documents, use one of these methods:

Method 1

1. Press **Help** until you reach the main help menu.
2. Tab to "Documents in the MAPPER Library."
3. Transmit.

Method 2

1. After signing on to the MAPPER system, move the cursor to the control line.
2. Type `help,doc`.
3. Transmit.

For a List of New Features

To obtain a list of new features for this release of MAPPER software, use one of these methods:

Method 1

1. Press **Help** until you reach the main help menu.
2. Tab to "New Features and Enhancements for this Release."
3. Transmit.

Method 2

1. After signing on to the MAPPER system, move the cursor to the control line.
2. Type `help,new`.
3. Transmit.

Contents

Section 3. MAPPER Function Finder

Section 4. MAPPER Functions and Runs

Appendix C. Using Your Keyboard and Mouse

Appendix D. Control Commands for Transferring Data

Appendix E. Mathematical Operators and Functions

Appendix F. Application Power Tools

Figures

Tables

Section 1
Introduction

You may find additional information specific to your MAPPER system by pressing **ReadMe** from the sign-on screen.

This manual describes functions for the MAPPER system. Use this manual to find out how to use options and parameters for the manual functions. An overview of MAPPER software is included as a refresher — if you have never used MAPPER software before, read the *Manual Functions Training Guide*. This section explains what MAPPER software is and defines what a manual function is.

What Is MAPPER Software?

MAPPER software is real-time report processing software. Real-time processing implies the recording of events as they occur — the immediate updating and analysis of reports. Data is stored in sets of columns of information called reports.

You do not have to understand programming to understand and use MAPPER software. Whatever your level of computer use, the MAPPER system is designed for you.

Definitlon of a Manual Function

A manual function is a component of MAPPER software that allows you to display and revise the information stored in your database, as well as manipulate and manage reports or results. There are over 80 manual functions available in this version of MAPPER software.

For example, if you used MAPPER software for your freight-handling business, you could display a report and use the Change (CHG) function to replace all occurrences of the character string Freight01 with Freight02.

Section 2 gives an overview of MAPPER software structure. Section 4 lists and describes several of the manual functions. Those not described in Section 4 can be found in the online help system (HELP,*call*).

Definition of a MAPPER Run

With a MAPPER run you can perform a series of functions repetitively. How? You create a single report and type in it a series of run statements. Each run statement performs a different function, such as subtotaling a field of data or searching for a character string. Whenever you want to perform that series of functions, you execute the run by entering a single command — the MAPPER run name, the name given to the sequence of run statements.

For example, if you frequently must find subtotals of a certain report field and then search the entire database for a given character string, you can automate this chore by writing a run and giving the run a name. When you enter the run name on the control line, the series of functions is performed automatically.

Some runs have been written and are ready for you to use. Some of these runs are listed with the manual functions in Section 4 of this manual.

To learn about writing MAPPER runs, refer to the *Run Design Training Guide* and the *Run Design Reference.*

MAPPER System Coordinators

MAPPER system coordinators control the development and use of the MAPPER database and are responsible for MAPPER security. They also plan, prepare, and administer demonstration and training exercises for users, and they coordinate database use and development with other support personnel.

Section 2
MAPPER Software: An Overview

You may find additional information specific to your MAPPER system by pressing **ReadMe** from the sign-on screen.

This section contains introductory information about your MAPPER system and covers the following topics:

- MAPPER database structure
- Execution of MAPPER software
- MAPPER reports and screens
- Execution of MAPPER functions
- MAPPER security features

MAPPER Database Structure

The method of organizing information within the MAPPER database is similar to that of a traditional office. Think of your MAPPER database as a large storage room. Within this room are filing cabinets containing drawers. These drawers, in turn, contain information in the form of reports.

The MAPPER database adapts this structure to an electronic database; even the terminology remains the same. You access cabinets, drawers, and reports to recall, review, and revise the information in your MAPPER database.

Cabinets

Sections of the database called cabinets (so named because they can be compared to filing cabinets) are used to store important information on a specific aspect of your organization or operation. For example, one cabinet might contain information on the accounting department for your organization, a second cabinet might be dedicated to the sales department, and a third to the shipping department.

Cabinets are numbered and matched in pairs that include an even number and an odd number, starting with cabinet pair 0/1. Each cabinet in the pair holds exactly the same data, but you can update information in the even-numbered cabinet, and only read the information in the odd-numbered cabinet. The system contains 210 cabinet pairs: cabinets 0 through 421.

Drawers

Just as a filing cabinet contains drawers, each MAPPER software cabinet contains several different drawers. Each drawer is a subcategory for the specific topic to which that cabinet is dedicated. For example, the cabinet for a sales department might contain one drawer for each salesperson.

Drawers are named by letter — A through I. Drawer A is the freeform drawer which is like a blank sheet of paper that can be used for any sort of data or communication. Drawer A is shared by all cabinets, but it is not displayed in the drawer table of contents. It contains general information, which is not protected.

The remaining drawers, B through I, are distinct databases, usually laid out in line and column format. They are planned by one or more users and generated by the coordinator.

Reports

Finally, each drawer can be used to store a large number of reports containing detailed information — for example, all sales closed during a specific month for a particular salesperson.

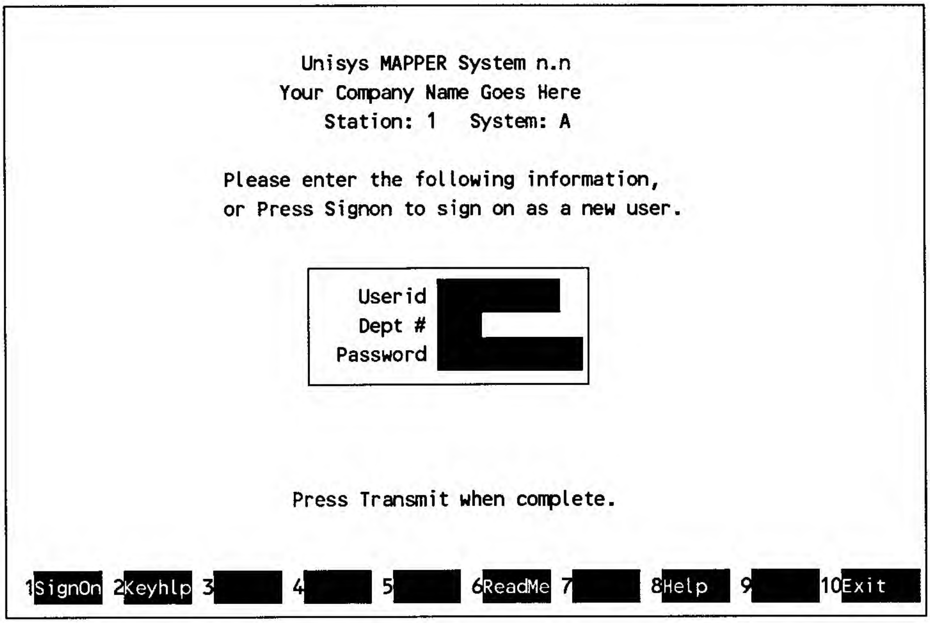
Each report is identified by a report name; or report number and drawer letter; or report number, drawer letter, and cabinet number. See the NAME run for information on how to identify your reports with a substitute name. The report identifier is used to identify a report, for example, report 2 in drawer B is called report 2B, report 4 in drawer C is called report 4C.

When it is important also to distinguish the cabinet that contains the report, add the cabinet number. For example, report 2B in cabinet 0 is referred to as report 2B0.

Reports can contain a limited number of lines; see your coordinator for the exact number. For the most efficient use of system resources, a report size of 100 to 500 lines is recommended.

Execution of MAPPER Software

When you execute MAPPER software, the MAPPER sign-on screen is displayed. This screen indicates what level of MAPPER software you are using, what your station (terminal) identification number is, and what MAPPER site letter you are using. From here, you must enter a registered user-id, department number, and password (see your MAPPER system coordinator for these), or press the **SignOn** key to gain access to the MAPPER system with a user-id of NEWUSER.



```
                    Unisys MAPPER System n.n
                   Your Company Name Goes Here
                     Station: 1   System: A

                Please enter the following information,
                or Press Signon to sign on as a new user.

                         Userid
                         Dept #
                       Password


                   Press Transmit when complete.

1SignOn 2Keyhlp 3       4       5       6ReadMe 7       8Help   9       10Exit
```

When you are done using the MAPPER software, you must sign off and exit MAPPER software. Refer to the Sign off MAPPER Software (X) and Exit MAPPER System (EXIT) functions.

A Series: See your coordinator for the specific logon instructions for your system.

Using the U Series MAPPER System and the UNIX MAPPER Systems

This section about executing MAPPER software applies only to the U Series MAPPER System and the UNIX MAPPER Systems.

You can execute MAPPER software that is installed on a UNIX system using either of these methods:

- Use the `mapper` login.
- Log in to the UNIX operating system and execute MAPPER software as a program.

Since logins and options are case sensitive on UNIX, you must enter these commands in lowercase to execute MAPPER software.

MAPPER Software Login

The easiest way to execute MAPPER software is with the `mapper` login. You enter this from the UNIX login prompt to execute software for the default MAPPER site. (The MAPPER default site on any host computer is the site that was installed first.)

The `mapper` login does not require a password. Simply type `mapper` and press **Return**. After a few seconds, the MAPPER sign-on screen is displayed and you can sign on.

You do not need a UNIX login to execute MAPPER software with this method; however, you can only access the default site. To access another site, a UNIX login is needed to allow you to execute MAPPER software as a program.

Executing MAPPER Software as a Program

When executing MAPPER software as a program, you can specify one or more of the following options:

- A specific MAPPER site
- A user-id with which to sign on and, optionally, a run name to execute
- A specific station identification or any available station to access

First log in to UNIX with your UNIX login. When the shell prompt is displayed, use the following format to type the command. Press **Return** to execute the MAPPER software.

```
mapper [-ssl -luser-id,dept,psw,run,data1{,data2,...}Δ
{-t|-Tdevice-name}]
```

where:

-ssl Site option, -s, and the letter of the MAPPER site to access, *sl*.

-l Indication that a MAPPER sign-on follows. (This l, letter "el," must be lowercase.)

user-id User-id with which to sign on to MAPPER system.

dept Department number.

psw Optional password. If no password is required and a run name is given, the commas designating the password field must be provided.

run Run name to execute after sign-on.

data1,data2,... Literal data to pass to the run.

-t Indicator to use the first available station number that is not configured without checking the configuration report. (This t must be lowercase.)

-T*device-name* Indicator to use the station number configured as *device-name*. (This T must be uppercase.)

Example 1: Executing MAPPER Software at Site C

User NEWUSER signs on to department 7 of site C and the HELP run executes:

```
$ mapper -sc -lnewuser,7,,help
```

Example 2: Executing MAPPER Software at a Specific Station

This command configures the station number tty08:

```
$ mapper -sc -Ttty08
```

Using the BTOS II MAPPER System: MAPPER ACCESS Command

This section about executing MAPPER software applies only to the BTOS II MAPPER System.

With the BTOS II MAPPER system, you use the MAPPER ACCESS command at the Executive command line to begin a MAPPER work session. You execute the MAPPER ACCESS command at a master or cluster BTOS workstation.

1. Type **mapper access** in the Executive command line.
2. If you need to specify any MAPPER ACCESS command parameters, press **Return** and type the parameter information in the command form fields.

Parameter	Description
B-Net node	Connects cluster to other clusters through the BTOS networking system. Examples of assigned values are node A or node WS001
Graphics device	Device available in the cluster or neighboring cluster through B-Net. Default value = none.
Color graphics?	Enter **y** for color graphics device, **n** for monochrome. Laser output is always designated N. Default value = N.

3. Press **GO** after specifying the MAPPER ACCESS command parameters, or to execute the command with the default parameters.
4. The MAPPER sign-on screen is displayed. Enter your MAPPER sign-on to begin your work session.

MAPPER Reports and Screens

Each report is a part of the total MAPPER database. MAPPER functions process the data stored in the reports.

The number of reports in a drawer varies. Usually, each report has fewer than 500 lines. The MAPPER system coordinator controls the size of reports and drawers and assigns new drawers for new applications. MAPPER software works most efficiently with a database of many short reports.

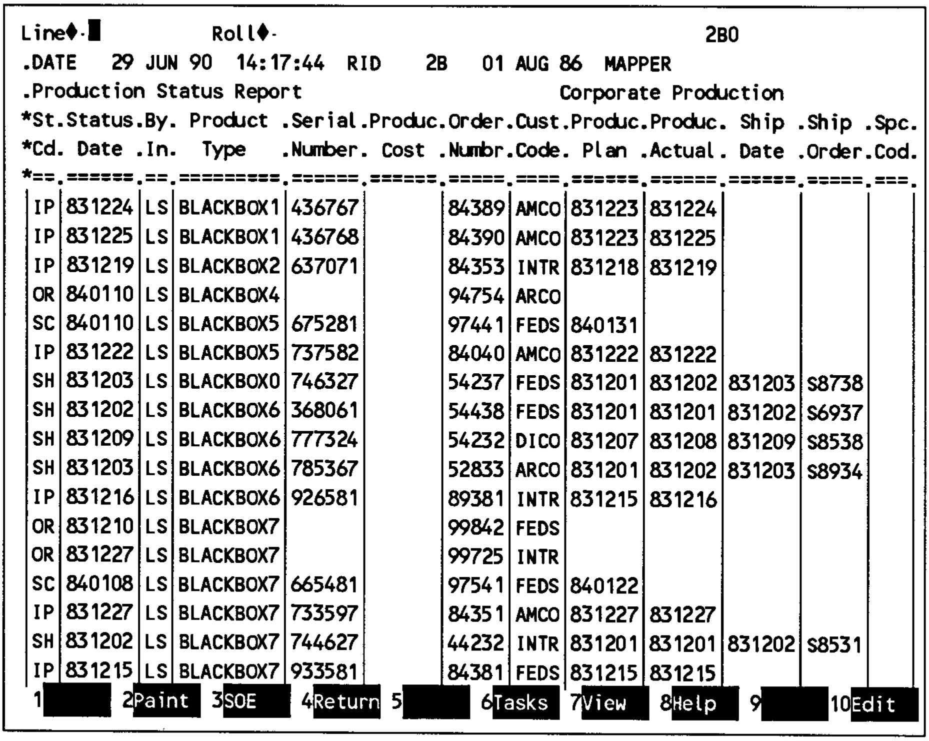
You can view the data in reports anytime on your display screen, and you can position it in a number of ways vertically and horizontally.

Choosing specific formats allow you to view a specific selection of report fields. There can be up to 25 report formats within each drawer. You can process reports displayed in any of these formats. For example, if format 2 is defined as the first, third, and fourth report fields, then when you request a report in that format, only those fields are displayed. The basic format, or format 0, is predefined as the leftmost columns that fit on your screen. But your coordinator can define formats 1 through 25, depending on your needs.

1100: There are 6 formats on the OS 1100 MAPPER System; you can define formats 1 through 6.

Report Screens

When you display a MAPPER report, a screen similar to the following is displayed:

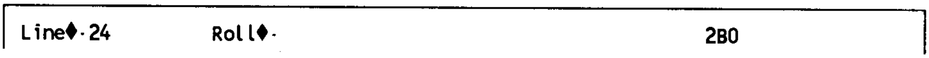


```
Line♦·█          Roll♦·                                        2B0
.DATE   29 JUN 90  14:17:44  RID    2B   01 AUG 86  MAPPER
.Production Status Report                      Corporate Production
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.Produc. Ship .Ship .Spc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .Actual. Date .Order.Cod.
*==.======.==.=========.======.======.=====.====.======.======.======.=====.===.
 IP 831224 LS BLACKBOX1 436767        84389 AMCO 831223 831224
 IP 831225 LS BLACKBOX1 436768        84390 AMCO 831223 831225
 IP 831219 LS BLACKBOX2 637071        84353 INTR 831218 831219
 OR 840110 LS BLACKBOX4               94754 ARCO
 SC 840110 LS BLACKBOX5 675281        97441 FEDS 840131
 IP 831222 LS BLACKBOX5 737582        84040 AMCO 831222 831222
 SH 831203 LS BLACKBOX0 746327        54237 FEDS 831201 831202 831203 S8738
 SH 831202 LS BLACKBOX6 368061        54438 FEDS 831201 831201 831202 S6937
 SH 831209 LS BLACKBOX6 777324        54232 DICO 831207 831208 831209 S8538
 SH 831203 LS BLACKBOX6 785367        52833 ARCO 831201 831202 831203 S8934
 IP 831216 LS BLACKBOX6 926581        89381 INTR 831215 831216
 OR 831210 LS BLACKBOX7               99842 FEDS
 OR 831227 LS BLACKBOX7               99725 INTR
 SC 840108 LS BLACKBOX7 665481        97541 FEDS 840122
 IP 831227 LS BLACKBOX7 733597        84351 AMCO 831227 831227
 SH 831202 LS BLACKBOX7 744627        44232 INTR 831201 831201 831202 S8531
 IP 831215 LS BLACKBOX7 933581        84381 FEDS 831215 831215
 1       2Paint  3SOE    4Return 5       6Tasks  7View   8Help   9       10Edit
```

Control Line and Line 1

The first line on the screen is the control line. The control line has two fields that allow you to display different parts of the report and one field (on the far right) that denotes the report being processed. Function calls are also entered on the control line.

Here is a typical control line:



```
Line♦·24         Roll♦·                                        2B0
```

The Line field indicates the number of the line currently at the top of the screen. The preceding control line indicates that line 24 of the report is at the top of the screen. To bring a different line to the top, type its number in the Line field. For example to display line 15 at the top of the screen, type **15** in the Line field.

Use the Roll field to scroll the display forward or back. To scroll forward, type + or leave the field empty and transmit. To scroll back, type - and transmit. To scroll forward a certain number of lines, type + and the number of lines, then transmit. To scroll back a certain number of lines, type - and the number of lines, then transmit. For example, **-13** scrolls the display back 13 lines.

You can shift the displayed report in other ways too. For example, using the Display Alternate Format (F*n*) function you can display a different selection of fields. For more information, see the following functions in the online help system: Create Temporary Format (VIEW), Display Alternate Format (F*n*), Display and Hold Headings (DH), Hold Lines on Screen (H*n*), Shift Display (S*n*).

When a report is on display, the first line of a report is the date line, which is also called line 1. If you are signed on to the demonstration database in cabinet 0, tab to the right of an SOE character (♦) on the control line and enter **2b**.

The date line, or line 1, looks like this:

```
.DATE   29 JUN 90  14:17:44  RID    2B   01 AUG 86  MAPPER
```

Following is an explanation of each field of the date line:

.DATE:	Date line identifier (also called line 1)
29 JUN 90:	Date of last update
14:17:44:	Time of last update
RID 2B:	Report number and drawer letter
01 AUG 86:	Date of origin of report or date report was last replaced
MAPPER:	User-id of last user to update report

Line Types

MAPPER reports can contain different types of lines, or line types, which are used to distinguish different kinds of data. For example, the line type determines whether the line is column-formatted or whether the system edits it for allowable characters. The line type is defined by the character in column 1.

If you are designing a report, use line types to specify how the system will interpret your data. Use the following descriptions to decide which line types to use.

If you are using a report that someone else created, read the following descriptions of each line type to understand how the report was set up and how to use each type of line.

The leftmost character on a line determines the line type. MAPPER reports can have these kinds of line types:

- Tab — Column-formatted edited lines, starting with a tab character (·)
- Asterisk — Column-formatted nonedited lines, starting with an asterisk (*)
- Period — Freeform comment lines, starting with a period (.) (If you look back at line 1, you can see that it is a period line.)
- Special — Column-formatted edited lines, starting with any valid character but not a tab, asterisk, or period

Note: *When determining which line types to use, it is important to note that asterisk and period lines are trailer lines. This means that they are associated with the previous data line and remain with the data line when it is processed. For example, a Search result shows all tab lines found, along with any asterisk or period lines that follow each tab line.*

Asterisk lines are trailer lines to tab lines, and period lines are trailer lines to asterisk lines.

Tab Lines

Type a tab character in column 1 to specify a column-formatted, edited line. The tab character is commonly represented on the screen by a vertical bar (|) but may be displayed as a space or a small dot. Most data lines in MAPPER reports are tab lines.

Tab lines range from 40 to 256 characters in length. You can display and process tab lines in different formats. Most data lines in a report are tab lines.

Asterisk Lines

Type an asterisk in column 1 to specify a column-formatted, nonedited line. Use asterisk lines for report headings. You can also use asterisk lines for data lines. The Iterative Calculate (ICAL) run, for example, uses asterisk lines to store equations.

Asterisk lines range from 40 to 256 characters in length and can be displayed and processed in different formats.

Period Lines

Type a period in column 1 to specify a freeform, nonedited line. The date line and title line of MAPPER reports are period lines. You may also use period lines to include comments in the report, since MAPPER software does not interpret these lines as data. Period lines do not shift from left to right when you manipulate the report display.

Period lines are restricted to 80 to 132 characters in length. However, since period lines do not shift, if your terminal is only 80 characters wide, your will only see 80 characters of the data.

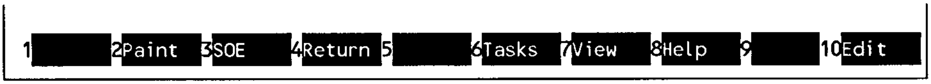
Period lines are considered trailer lines if they follow asterisk or tab lines.

Special Lines

Type any character other than a tab, period, or asterisk in column 1 to specify a column-formatted, edited line. Special lines have all the other characteristics of tab lines.

Function Key Bars

A function key is displayed at the bottom of the screen and contains labels describing the operation performed by each function key.



For example, if you need information about the MAPPER system or a specific function, press the **Help** key, which in this case is **F8**. Or if you need to print a report, but do not know which function to use, press **Tasks**. This key displays a menu from which you can choose the task you want to perform, such as finding data, comparing reports, or printing reports.

Note that the function keys included in the function key bar vary depending on what you are doing. Also note that applications at your site may change the function key definitions.

See Appendix C for a description of what each function key does.

The Demonstration Database

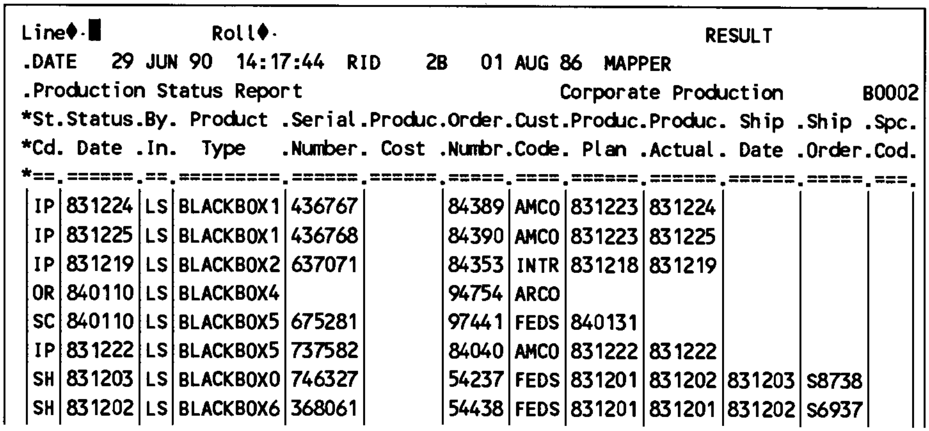
The demonstration database consists of reports on which you can practice functions.

Caution

Before making changes to a standard report in the demonstration database, duplicate the report using the Duplicate Report (XR) function. This way, you will not alter or delete part of this database for succeeding users.

Results

When you perform certain MAPPER functions, a result is displayed showing the outcome of the operation. The word RESULT is displayed at the far right of the control line.



```
Line♦·█          Roll♦·                                         RESULT
.DATE   29 JUN 90  14:17:44  RID    2B   01 AUG 86  MAPPER
.Production Status Report                       Corporate Production     B0002
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.Produc. Ship .Ship .Spc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .Actual. Date .Order.Cod.
*==.======.==.=========.======.======.=====.====.======.======.======.=====.===.
 IP|831224|LS|BLACKBOX1|436767|      |84389|AMCO|831223|831224|      |     |   |
 IP|831225|LS|BLACKBOX1|436768|      |84390|AMCO|831223|831225|      |     |   |
 IP|831219|LS|BLACKBOX2|637071|      |84353|INTR|831218|831219|      |     |   |
 OR|840110|LS|BLACKBOX4|      |      |94754|ARCO|      |      |      |     |   |
 SC|840110|LS|BLACKBOX5|675281|      |97441|FEDS|840131|      |      |     |   |
 IP|831222|LS|BLACKBOX5|737582|      |84040|AMCO|831222|831222|      |     |   |
 SH|831203|LS|BLACKBOX0|746327|      |54237|FEDS|831201|831202|831203|S8738|   |
 SH|831202|LS|BLACKBOX6|368061|      |54438|FEDS|831201|831201|831202|S6937|   |
```

A result is a temporary report produced after a MAPPER function completes processing. This result is lost as soon as you display another report, exit the report, or sign off. If you want to keep the result, you can do one of the following:

- Replace the result into a permanent report using the Replace Report (REP) function
- Duplicate the result into a permanent report using the Duplicate Report (XR) function

In addition to printing, duplicating, and replacing results, you can perform many of the other functions and runs described in this manual. To execute a function or run against a result on display, use a minus sign (-) in place of a report number.

Execution of MAPPER Functions

You can request a MAPPER function in one of two ways: proceeding through a menu path or entering a command, also called a function call, on the control line. Once you have requested a function, you may have to complete a function form and function mask in order for the function to complete its execution. (See also "Function Forms" and "Function Masks" later in this section.)

Proceeding through a Menu Path

You can access most MAPPER functions through menu paths. For example, follow this procedure to request the Totalize (TOT) function through a menu path:

1. Press the **Task** key.

   Outcome: The Select Task menu is displayed.

2. Move the cursor to the **Perform math operations** selection and transmit.

   Outcome: The Perform Math Operations menu is displayed.

3. Move the cursor to the **Totalize data** selection and transmit.

   Possible outcomes:

   - If a report was on display, the function mask is displayed.
   - If a report was not on display, the function form is displayed.

   In some cases neither a function form nor a function mask are displayed. See "Why Function Forms Are Displayed" later in this section for more details.

The following text is notation used in this book for the above procedure:

```
Select Task
    Perform Math Operations
        Totalize Data
```

Entering a Command on the Control Line

You can request a MAPPER function by entering a command, or function call, on the control line. For example, follow this procedure to request the TOT function on the control line:

1. Place the cursor in the home position or in any control position in the control line.
2. Type `tot`.
3. Erase the rest of the line by pressing the **Erase to End of Line** key and transmit.

   Possible outcomes:

   - If a report was on display, the function mask is displayed.
   - If a report was not on display, the function form is displayed.

   In some cases neither a function form nor a function mask are displayed. See "Why Function Forms Are Displayed" later in this section for more details.

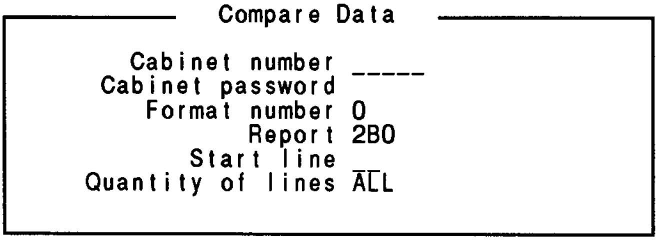
Many functions have more than one command format. Each command format serves a different purpose. For example, if you want to perform the TOT function on report 2B in cabinet 0, and that report is not on display, you can enter `tot 2b0` rather than `tot`.

*Note: If an system message erases the control line, press **Home** and enter* `L` *to restore it. See the L (Line Control) function. You can also press the **Paint** key if it is present on the function key bar.*

Function Forms

A function form is a screen that is displayed when you request a function. The form contains fields in which you specify how you want to use the function. Some fields are automatically filled with default values.

This function form is displayed when you request the Compare Data (CMP) function. Note that if a report was on display when you requested the function, some fields are automatically filled in with default values. You can use these values or change them by typing over them.



```
                 Compare Data

       Cabinet number _____
     Cabinet password
        Format number 0
               Report 2B0
           Start line _
    Quantity of lines ALL
```

Many functions, however, allow you to skip the function form and use a function format. This format, which you enter on the control line, lets you specify many of the same values as does the function form.

Procedure for Filling out a Function Form

After you access a function and the form is displayed, follow this procedure to complete the form:

1. Type information in each field you want to fill in or change.
2. When you have completed the function form, transmit.

Note: *Some fields require a value. If you leave a required field blank, a system message is displayed.*

Available Help

The following help is available for you when filling in a function form:

- For an explanation of a field and its possible values, tab to that field and press **Help**.

  Outcome: A brief description is displayed.

- For additional information about how the function works, press **Help** again.

  Outcome: The help topic for the function is displayed.

Screens Displayed After the Function Form Is Complete

After you complete a function form, the next screen displayed varies depending on the function and what was on display.

- Some functions execute immediately when you transmit the completed function form.
- Some functions require more information about the report fields to process. In these cases, a function mask is displayed.
- If the system cannot execute the function based on the information you supplied, it displays a system message. For example, if you did not fill in a required field or you specified a report that does not exist, the system displays a message and does not execute the function.

Why Function Forms Are Displayed

Function forms are displayed because the system needs more information to process your request. A function form may be displayed if one of the following is true:

- You request a function while no report is on display.
- You request a function that needs more information than just the report name and format number.
- You have a report on display, you access the function by entering only the function call (for example, `ar` rather than `ar b0`), and you use a function that can process an entire drawer.

Function Masks

A function mask looks like a report with blank fields. You use it to specify options, field widths, and parameters whenever you request a function that processes individual fields or character positions in a report. The field headings displayed in the mask are those that were displayed when you requested the function. If you process an entire drawer, the system displays headings from report 0.

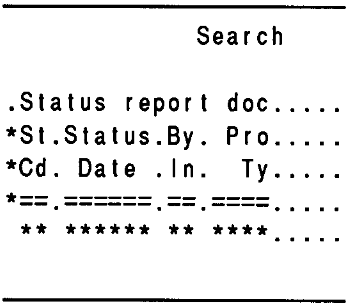
- **Options** are usually designated by single letters or symbols. You use them to change the way a function works. Type options in the line just above the report headings from left to right beginning in column one.
- **Field widths** are denoted by a line of asterisks below the heading divider line, making it easier to select the fields to process when entering function parameters.
- **Parameters** indicate which operations to perform and the order in which to perform them. They also let you select the types of lines to process.

Where to Type Options and Parameters

The descriptions and arrows to the left of this partial function mask point to the places where you type options and parameters as well as where you change the field width, if necessary.

Type options above the mask. -------------->

Change field width if necessary. ------------>
Type parameters below the mask. ---------->



```
                 Search

.Status report doc.....
*St.Status.By. Pro.....
*Cd. Date .In.  Ty.....
*==.=======.==.====.....
 ** ******* ** ****.....
```

Available Help

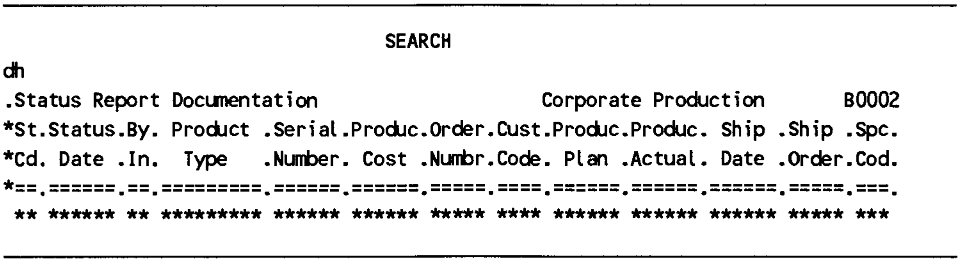
Whenever you have a function mask on display, you can press **Help** to see a summary of available options and parameters for the function you are using. Press **Help** again for more details on the function.

Specifying Options

Give more detailed instructions about how to process your request by using options. Since options vary by function, see specific functions for available options.

After you have requested a function and the function mask is displayed, type one or more options on the blank line just above the heading line in the mask. When you use more than one option, type them next to each other with no intervening spaces or commas.

Two options, D and H, have been typed in this function mask:



```
                                        SEARCH
dh
.Status Report Documentation                     Corporate Production      B0002
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.Produc. Ship .Ship .Spc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .Actual. Date .Order.Cod.
*==.======.==.=========.======.======.=====.====.======.======.======.=====.===.
 ** ****** ** ********* ****** ****** ***** **** ****** ****** ****** ***** ***
```

Specifying Parameters

Use parameters to indicate which fields to process and how to process them. Since parameters vary by function, see specific functions for available parameters.

After you have requested the function and the function mask is displayed, follow this procedure to type parameters on the function mask:

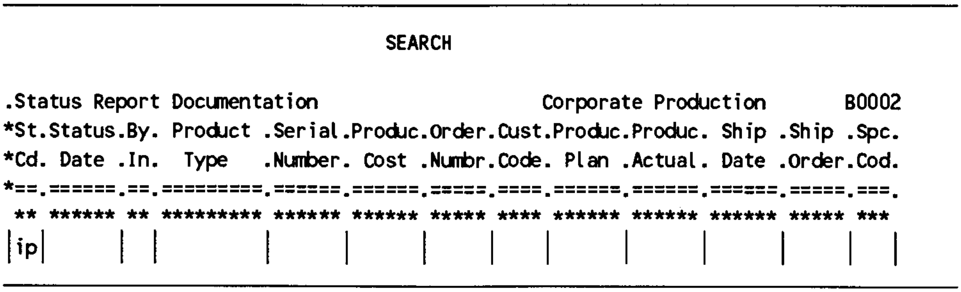
1. Tab to the field below the asterisk line where you want to enter a parameter.

2. Type the parameter.

3. If more parameters are necessary, repeat steps 1 and 2.
4. Move the cursor to the next blank line and transmit.

For functions that locate data, a parameter would be the character string to locate. For functions that perform math operations, a parameter would be +, -, and so on.

The parameter IP has been typed in this function mask. When executed, this function mask searches for occurrences of IP in the St Cd field:



```
                                SEARCH

.Status Report Documentation                Corporate Production     B0002
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.Produc. Ship .Ship .Spc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .Actual. Date .Order.Cod.
*==.======.==.=========.======.======.=====.====.======.======.======.=====.===.
 ** ****** ** ********* ****** ****** ***** **** ****** ****** ****** ***** ***
|ip|      | |         |      |      |     |    |      |      |      |     |   |
```

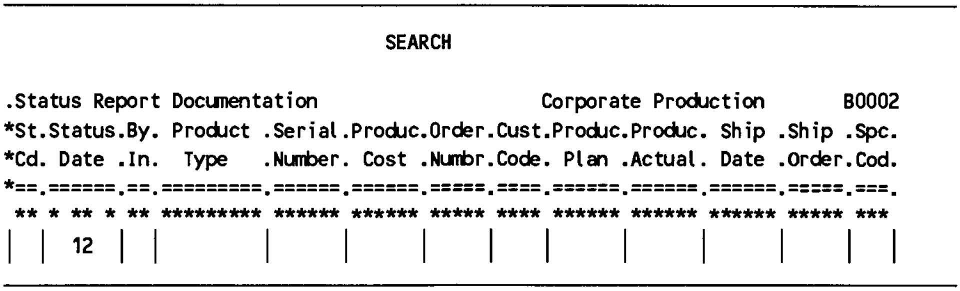
Processing Partial Fields

Sometimes you may need to process only part of a field. For example, you might need to search only the third and fourth columns of a six-column field. You can specify partial fields to be processed in two ways. The first method is useful for any function mask; the second is useful only for the functions that search reports.

Method 1 — For all functions. After you access a function, and the mask is displayed, follow this procedure:

1. Type spaces over, or erase, asterisks in the columns not to be processed. Note that you only need to erase the asterisks before and after the columns that are to be processed.
2. Type parameters below the appropriate asterisks.

In the following function mask, the second and fifth asterisks of the Status Date field are erased. This means that the third and fourth characters are to be processed. The parameter is typed below the asterisks of the columns to be processed.

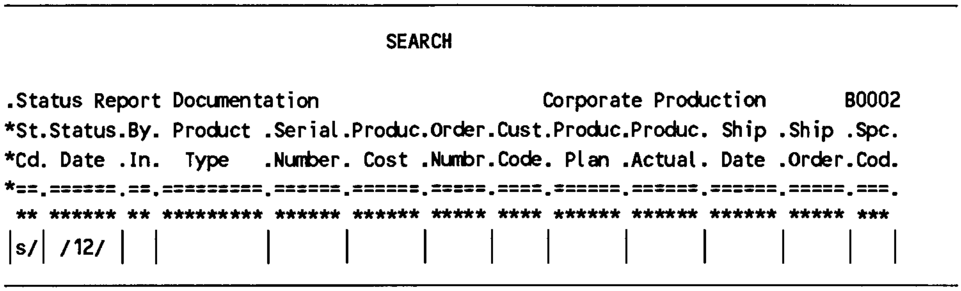


```
                                  SEARCH

.Status Report Documentation                   Corporate Production       B0002
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.Produc. Ship .Ship .Spc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .Actual. Date .Order.Cod.
*==.======.==.=========.======.======.=====.====.======.======.======.=====.===.
 ** * ** * ** ********* ****** ****** ***** **** ****** ****** ****** ***** ***
|  |  12  |  |         |      |      |     |    |      |      |      |     |   |
```

Method 2 — For functions that search data: After you access a function, and the mask is displayed, follow this procedure:

1. Type a slant (/) in the parameter line preceding the column where you want the function to start processing in the field. You can also type a slant following the last column to be processed in the field.
2. Type the string to locate in the remaining portion of the field.

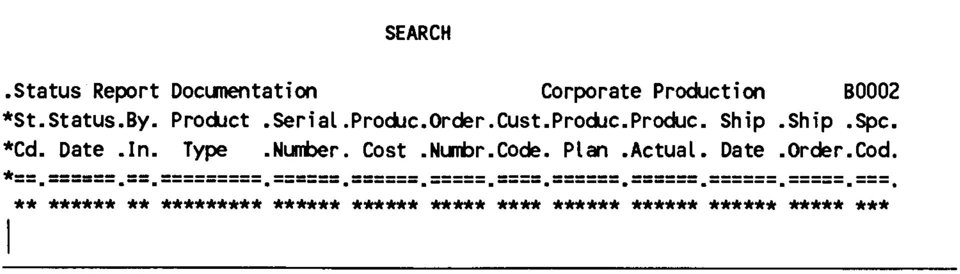
Here, in the St Cd field, column 1 is processed; in the Status Date field, columns 3 and 4 are processed:



```
                                  SEARCH

.Status Report Documentation                   Corporate Production       B0002
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.Produc. Ship .Ship .Spc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .Actual. Date .Order.Cod.
*==.======.==.=========.======.======.=====.====.======.======.======.=====.===.
 ** ****** ** ********* ****** ****** ***** **** ****** ****** ****** ***** ***
|s/| /12/ |  |         |      |      |     |    |      |      |      |     |   |
```

Selecting the Type of Lines to Process

Indicate the type of lines to process in the first column of the line under the asterisk line. For example, in this function mask, since the first character below the asterisk line is a tab (represented here by a vertical bar), the function processes tab lines only.



```
                                SEARCH

.Status Report Documentation                  Corporate Production      B0002
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.Produc. Ship .Ship .Spc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .Actual. Date .Order.Cod.
*==.======.==.=========.======.======.=====.====.======.======.======.=====.===.
 ** ****** ** ********* ****** ****** ***** **** ****** ****** ****** ***** ***
|
```

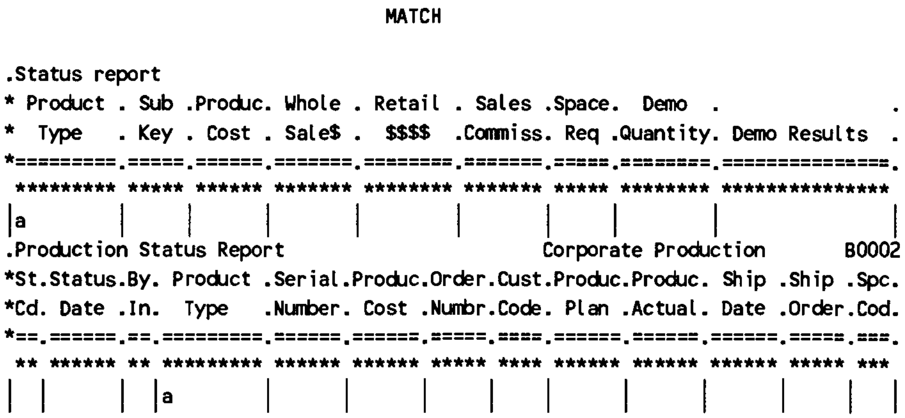
To change the type of line to process, type over the line-type character in column 1 with the character that represents the type of line to be processed.

For example, to process asterisk lines rather than tab lines, type an asterisk in column 1 of the line below the asterisk line. Some functions have options that change the type of lines to process. See individual functions for details.

Using Double Function Masks

Functions that process two reports at a time display two function masks on the screen.

The top mask is from the report you requested as a second report to process, referred to by some functions as the issuing report. The lower mask is from the current report, referred to by some functions as the receiving report.



```
                                 MATCH

.Status report
* Product . Sub .Produc. Whole . Retail . Sales .Space.  Demo  .               .
*  Type   . Key . Cost . Sale$ .  $$$$  .Commiss. Req .Quantity. Demo Results  .
*=========.=====.======.=======.========.=======.=====.========.===============.
 ********* ***** ****** ******* ******** ******* ***** ******** ***************
|a        |     |      |       |        |       |     |        |               |
.Production Status Report                       Corporate Production       B0002
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.Produc. Ship .Ship .Spc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .Actual. Date .Order.Cod.
*==.======.==.=========.======.======.=====.====.======.======.======.=====.===.
 ** ****** ** ********* ****** ****** ***** **** ****** ****** ****** ***** ***
|  |      |  |a        |      |      |     |    |      |      |      |     |   |
```

To fill in the double function mask, enter function options above the top mask, and parameters below each mask according to the function you are using. See individual functions for more details.

Fields Displayed in the Function Mask

By default, function masks display from left to right the field headings of the report you requested to process.

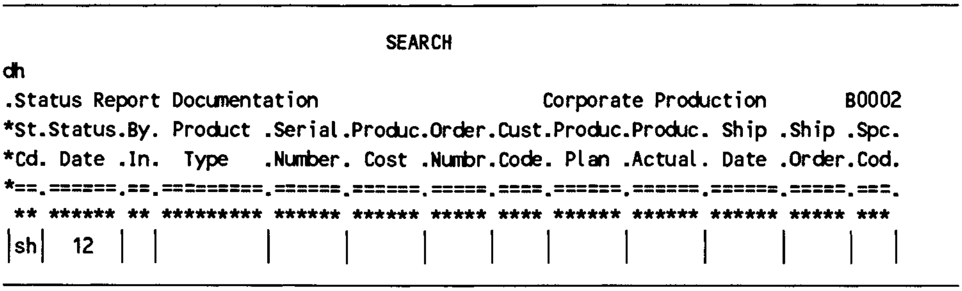
You can choose a different selection of fields in the following ways:

- Before requesting the function, display the report in a different format or change the user view of the report. Formats and views are variations on the way the report is displayed. You can display the report in another format by pressing **View** and using the menu, or by using the Display Alternate Format (F*n*) function to switch to a specified format.

- When you initially request the function, indicate another format either in the function call format or in the function form. See individual functions for more information.

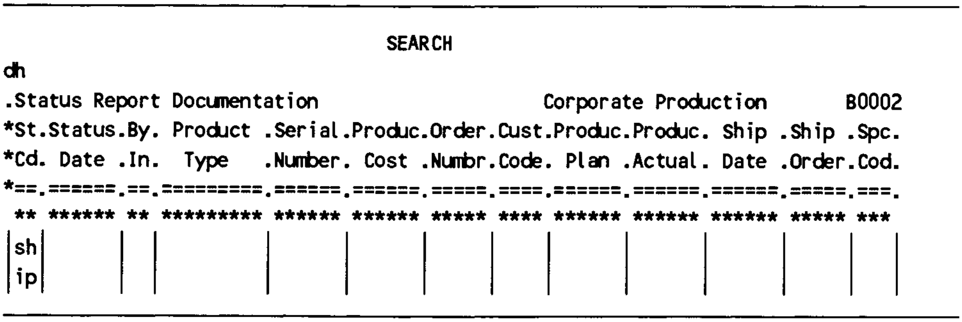
Example of Searching for Multiple Character Strings

With some functions, you can look for multiple strings. In this function mask, data lines are found containing SH in the St Cd field and a 12 in the Status Date field:



```
                                   SEARCH
dh
.Status Report Documentation                    Corporate Production      B0002
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.Produc. Ship .Ship .Spc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .Actual. Date .Order.Cod.
*==.======.==.=========.======.======.=====.====.======.======.======.=====.===.
 ** ****** ** ********* ****** ****** ***** **** ****** ****** ****** ***** ***
|sh|  12  |  |         |      |      |     |    |      |      |      |     |   |
```

In this function mask, data lines containing SH or IP in the St Cd field are found:



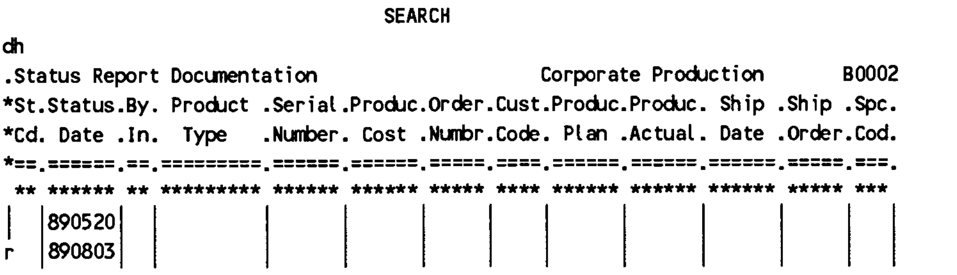
```
                                   SEARCH
dh
.Status Report Documentation                    Corporate Production      B0002
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.Produc. Ship .Ship .Spc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .Actual. Date .Order.Cod.
*==.======.==.=========.======.======.=====.====.======.======.======.=====.===.
 ** ****** ** ********* ****** ****** ***** **** ****** ****** ****** ***** ***
|sh|      |  |         |      |      |     |    |      |      |      |     |   |
|ip|      |  |         |      |      |     |    |      |      |      |     |   |
```

Example of Searching for a Range of Strings

Some functions let you search for a range of character strings. See the individual searching functions to see whether range searches are available. To search for a range of strings, access the Search function, display the function mask, and follow this procedure:

1. Type the characters for the lower end of the range in the first line under the mask in the correct field.
2. Type r in the first column of the next line.
3. In the same line as the R parameter, type the characters for the higher end of the range in the same field as the lower end. Then transmit.

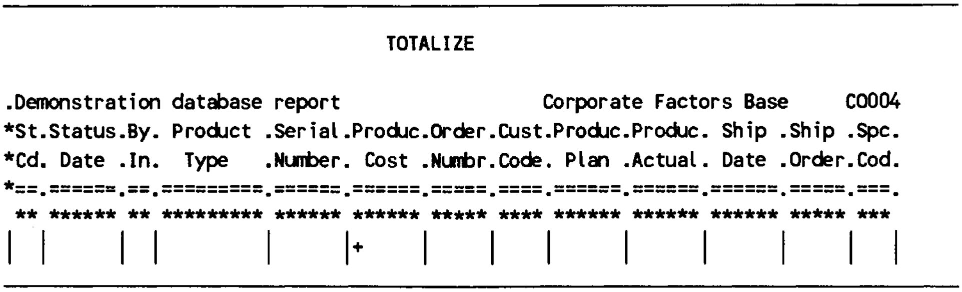
In this function mask, 890520 is at the lower end of the range, 890803 is at the higher end. Note that the R parameter is entered in column 1.



```
                                SEARCH
dh
.Status Report Documentation                  Corporate Production      B0002
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.Produc. Ship .Ship .Spc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .Actual. Date .Order.Cod.
*==.======.==.=========.======.======.=====.====.======.======.======.=====.===.
 ** ****** ** ********* ****** ****** ***** **** ****** ****** ****** ***** ***
|  |890520|  |         |      |      |     |    |      |      |      |     |   |
r  |890803|  |         |      |      |     |    |      |      |      |     |   |
```

Example of Using Mathematics

Some functions use arithmetic operators for parameters. In this Totalize function mask, the plus sign vertically adds the values in the Produc Cost field:

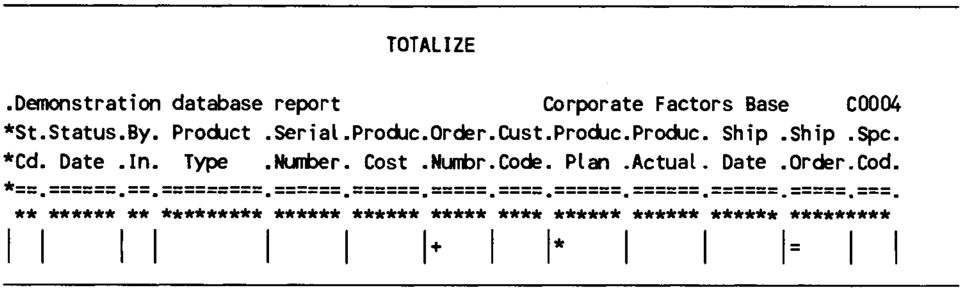


```
                                   TOTALIZE

.Demonstration database report                  Corporate Factors Base     C0004
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.Produc. Ship .Ship .Spc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .Actual. Date .Order.Cod.
*==.======.==.=========.======.======.=====.====.======.======.======.=====.===.
 ** ****** ** ********* ****** ****** ***** **** ****** ****** ****** ***** ***
|  |      |  |         |      |+     |     |    |      |      |      |     |   |
```

Example of Joining Fields in the Mask

To join two or more fields in a mask, type asterisks in the spaces in the line of asterisks.

This example temporarily joins two fields to create a field that is large enough to hold the result of a horizontal calculation. The fields become one field in the result, which you can print or duplicate (avoid replacing the original report because of the changed fields). Note the added asterisk between the Ship Order and Spc Cod.



```
                                   TOTALIZE

.Demonstration database report                  Corporate Factors Base     C0004
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.Produc. Ship .Ship .Spc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .Actual. Date .Order.Cod.
*==.======.==.=========.======.======.=====.====.======.======.======.=====.===.
 ** ****** ** ********* ****** ****** ***** **** ****** ****** ****** *********
|  |      |  |         |      |      |+    |    |*     |      |      |=    |   |
```

MAPPER Security Features

To continue with the previous office comparison, a limited number of individuals might be given keys to certain filing cabinets. Similarly, access to specific MAPPER cabinets, drawers, and reports can be controlled.

The security of the MAPPER system depends on controlling access to information. The MAPPER system coordinator is responsible for all MAPPER security for your department and determines the security structure. If you believe that your access to specific MAPPER functions is too restricted or that certain sensitive material necessitates even greater security measures, consult your MAPPER coordinator.

The MAPPER system has several features designed to protect the database against unauthorized users. These features include the user-id, user password, cabinet password, drawer password, report password, and run registration.

User-id

The MAPPER system coordinator registers you as a MAPPER system user and assigns you a unique user-id. The user-id allows you to access a specific cabinet, also assigned by the system coordinator, when you sign on to MAPPER software. It also controls the range of operations you can perform.

User Passwords

The MAPPER system may or may not require you to enter a password each time you sign on. You can create, remove, or change your own password, as often as you like. For information on user passwords, see the *Manual Functions Training Guide*. Also, see the PSW (Password) function.

Cabinet Passwords

The MAPPER system coordinator assigns cabinet passwords. To access a cabinet other than the one assigned to your user-id you can use the Cabinet Switch (C) function and enter the cabinet password, or use the Cabinet Switch (CS) run. Your system coordinator can supply you with the password if you need to access another cabinet.

The coordinator may also register your userid to have access to drawers in other cabinets. If you are registered, you can access a report in another cabinet without having to switch to that cabinet.

Drawer Passwords

A drawer password prevents an unauthorized user from displaying or updating any report in the drawer. The system coordinator assigns and resets the drawer password. You must obtain the password from the coordinator and use the Drawer Password (DPW) function to gain access to the protected drawer. (For details, see the DPW function.)

Report Passwords

You can restrict read or write access to a report by creating a password with either the Read Password or Password function. You then must enter the password to display or update the restricted report. For details, see the RPSW (Read Password) or PSW (Write Password) functions.

Run Security

Run security affects users working in run design, an advanced MAPPER feature. The MAPPER system coordinator must register any MAPPER run before you can execute it. The coordinator can impose limits on the run, such as who can access it or limit the time period in which it can execute. See your coordinator for additional information, or refer to the *Run Design Training Guide* and the *Run Design Reference*.

Section 3
MAPPER Function Finder

You may find additional information specific to your MAPPER system by pressing **ReadMe** from the sign-on screen.

This section categorizes MAPPER functions and runs into types of operations.

Note: *Since this section encompasses several different kinds of MAPPER systems, some functions may not be available on your system.*

The section includes these topics:

- Signing on to and exiting the MAPPER system
- Getting online assistance
- Accessing a MAPPER report and getting information about reports
- Creating MAPPER reports
- Updating individual MAPPER reports
- Updating MAPPER reports from update results
- Finding and comparing specific MAPPER data
- Generating MAPPER results
- Performing mathematical operations
- Processing dates
- Saving and deleting MAPPER reports
- Storing and retrieving MAPPER reports offline
- Printing MAPPER reports
- Performing interactive MAPPER operations

Signing on to and Exiting the MAPPER System

Use the following functions to enter (sign on) and exit:

- Signing on to MAPPER software
  - Sign on MAPPER Software function
- Terminating an active MAPPER session
  - Sign off MAPPER Software (X) function
- Exiting the MAPPER system
  - Exit MAPPER System (EXIT) function

Getting Online Assistance

If you know how to sign on to the MAPPER system, you can learn a great deal about how to use it. The following functions and runs display information about the MAPPER system:

- Obtaining help
  - HELP run
- Displaying report and line limits
  - LIMITS (Display Report and Line Limits) run
- Displaying a line count on the screen
  - DLC (Display Line Count) run
- Changing language of MAPPER software messages
  - Language (LANG) function

Remember that you also can obtain online assistance by pressing the **Help** key.

Accessing a MAPPER Report and Getting Information About Reports

You view a report by displaying it on the MAPPER terminal. The functions that display information about the MAPPER reports are listed below:

- Selecting MAPPER cabinets
  - Cabinet Switch (C) function
- Displaying MAPPER drawers
  - Drawer Table of Contents (T) function
- Requesting a report index
  - Index (I) function
- Requesting a report index by user
  - Index User (IU) function
- Displaying a MAPPER report
  - Display Report (D) function
- Naming a MAPPER report
  - NAME run
- Displaying a list of named reports
  - NAMES run
- Unlocking a drawer
  - Drawer Password (DPW) function
- Displaying line 0 information
  - Line Zero (LZ) function

You can control how MAPPER reports are displayed on the terminal. The control line positions and functions used to change how reports are displayed are listed below:

- Adjusting a report display
  - Create Temporary Format (VIEW)
  - Display Alternate Format (F*n*)
  - Display and Hold Headings (DH)
  - Hold Lines on Screen (H*n*)
  - Hold Characters on Screen (HC*n*)
  - Line control position (Line♦)
  - Roll control position (Roll♦)
  - Shift Display (S*n*)
- Releasing a displayed report or result
  - Release Display (^) function
- Repainting your screen
  - Paint (PNT) function (also the **Paint** key)

Creating MAPPER Reports

These are the functions for creating reports:

- Adding reports to a drawer
  - Add Report (AR) function
- Duplicating reports within a drawer
  - Duplicate Report (XR) function
- Replacing reports within a drawer
  - Replace Report (REP) function

Updating Individual MAPPER Reports

You can revise the contents of a report displayed on your terminal. These functions do not create any form of result. The permanent report is immediately updated to reflect the changes.

The following functions and procedures outlined in this section affect only the displayed report:

- Updating data in report lines
  - SOE Update function
- Duplicating report lines
  - Duplicate Line function
- Moving lines within a report
  - Move Line function
- Inserting lines in a report
  - Insert Line function
- Adding new report lines
  - Add Line function
- Deleting unwanted report lines
  - Delete Line function
- Copying lines from a temporary yank or append buffer into a report or result
  - Put Line function
- Sorting report and replacing sorted result into the original report
  - Sort and Replace Report (SORTR) function

The following functions update the line 0 information with the password, but do not make a visible change to a report:

- Adding a write password to a report
  - Password (PSW) function
- Adding a read password to a report
  - Read Password (RPSW) function

The following functions create or add lines to temporary buffers to be used by the Put Line function:

- Copying lines from a report or result and appending the lines to an existing temporary buffer
  - Append Line function
- Copying lines from a report or result into a temporary buffer
  - Yank Line function

Updating MAPPER Reports from Update Results

Several functions create update results that allow changes to the update results to be integrated back into the original report.

These functions manipulate update results into original reports:

- Integrating update result lines into the original report
  - Update (UPD) function
- Deleting update result lines from the original report
  - Delete (DEL) function
- Deleting update result lines from the original report and redisplaying the result
  - Extract (EXT) function

Finding and Comparing Specific MAPPER Data

The following functions allow you to find or compare specific data stored within the MAPPER database:

- Locate specific items in sorted fields and display the report
  - Binary Find (BF) function
- Match and move data from selected fields in two different reports
  - Match (MA) function
- Scan a report, result, or an entire drawer for all occurrences of specified data
  - Search (S) function
- Locate and display character strings
  - Locate (LOC) function
- Find and display an item in a field
  - Find (F) function
- Compare data in two reports
  - Compare Report (CMP) function

Generating MAPPER Results

Certain functions do more than merely change the data on a current report. These functions actually create a new report called a result. Your original report is unchanged.

A result is not automatically saved when you exit that result. To save a result, use either the REP (Replace Report) or XR (Duplicate Report) functions.

These function calls and procedures generate results:

- Creating a result
  - Create Result Copy (RSLT) function
- Sorting reports
  - Sort function
- Performing search operations
  - Change (CHG) function
  - Search (S) function
  - Search Update (SU) function
  - Match (MA) function
  - Match Update (MAU) function
- Reformatting a report
  - Reformat Report (RF) function
- Appending reports
  - Adding on to the current report — Add On (ADON) function
  - Appending the current report to another report — Add To (ADTO) function
- Encoding a report
  - Encode Report (ENCODE) function
- Decoding an encoded report
  - Decode Report (DECODE) function

Performing Mathematical Operations

MAPPER mathematical operations enable you to calculate or totalize the information in your reports. Performing these functions generates a result. A comprehensive explanation of all mathematical operations available with the MAPPER system is found in Section 4.

Arithmetic Operations

Arithmetic operations include those that are performed using arithmetic expressions. The following operations are available under the Arithmetic (A) function in Section 4:

- Addition
- Subtraction
- Division
- Multiplication
- Exponentiation
- Conversion of positive to negative numbers
- Absolute values
- Integer values
- Fractional values
- Log values
- Square roots
- Cube roots
- Sines and cosines
- Tangents and cotangents
- Arc sines, cosines, and tangents
- Hyperbolic sines and cosines
- Radian tangents and cotangents
- Pi equations

Totalize Operations

The Totalize (TOT) function allows you to perform arithmetic or move operations on fields within reports or results. The TOT function creates a result. The following operations are available:

- Counting and numbering
- Addition
- Subtraction
- Multiplication
- Division
- Subtotals
- Cumulative totals
- Averaging
- Deleting or inserting commas in numeric fields
- Moving data from one field to another

Calculate Operations

The following operations are available under the Calculate (CAL) function:

- Conditional display of numeric fields
- Elimination of zero results
- Elimination of nonsignificant zeros
- Addition of leading zeros
- Setting of initial value of numeric entries
- Conversion of nonnumeric to numeric values
- Rounding of numeric values
- Flagging of invalid numeric results
- Automatic exclusion of invalid result fields
- Conversion of numeric fields to integer format

- Right-justification and formatting of numeric fields
- Left-justification and formatting of numeric fields
- Calculations with date or time variables

Calculate Update Function

The Calculate Update (CALU) function allows you to do the following:

- Create an update result and integrate the lines of the update back into the original report using the Update (UPD) function
- Delete the update result lines from the original report using the Delete (DEL) function
- Delete the lines of the update result from the original report and redisplay the result for further processing using the Extract (EXT) function

Count Function

The Count (CNT) function is one of the most powerful of the mathematical functions. The following list contains some of the tasks it allows you to do:

- Subtotal fields without having to sort them first
- Perform calculations on a single report or across an entire drawer
- Process more than one distinctly different set of analyses with one function request
- Reduce processing time because internally, the function scans the report in one pass no matter how many analyses you request. You also reduce processing time because in many cases, one Count request can take the place of a sequence of two, three, or even four other functions.
- Display only specified fields in the result
- Numerically round each requested field separately
- Perform calculations on intervals of data (scaling)

Processing Dates

The ICAL run, and the DATE, CAL, and CALU functions perform computations on dates recorded in your MAPPER reports.

You can use these functions and runs to perform many tasks:

- Compare dates in a report with the current date or with other dates within that report
- Alter the format of dates within a MAPPER report
- Change a date field by adding days to or subtracting days from the original date field

Saving and Deleting MAPPER Reports

As you make changes to a MAPPER report, you must ensure that the report reflects those changes. When you execute an update function, updates are automatically made to permanent MAPPER reports. However, as mentioned earlier, results are not permanent. To make a result permanent, you must save the result before accessing another MAPPER report or result.

As reports become outdated, you should delete them to free up valuable storage space.

These functions allow you to save MAPPER results and delete outdated MAPPER reports:

- Replacing reports within a drawer
  - Replace Report (REP) function
- Duplicating reports within a drawer
  - Duplicate Report (XR) function
- Deleting MAPPER reports
  - Delete Report (DR) function

Storing and Retrieving MAPPER Reports Offline

Reports are stored directly within a drawer in the MAPPER system database. In some cases, however, it is advantageous to store specific reports in an offline storage device other than your main MAPPER database. This section discusses the storage and retrieval of MAPPER reports from offline devices, such as native data files.

- Copying reports to a native data file
  - Create File (FILE) function
- Copying an OS 1100 program file, element, or data file from one site to another site through the remote run link
  - Copy (COPY) function
- Copying a MAPPER report to a standard OS 1100 program file or symbolic element, or to a data file
  - Element (ELT) function
- Deleting a standard OS 1100 program file or symbolic element, or data file
  - Element Delete (ELT-) function
- Retrieving reports from a native data file
  - Retrieve File (RET) function
- Retrieving OS 1100 data files or elements of program files
  - Retrieve (RET [P]) function
- Transferring reports between the MAPPER system and diskette drive
  - Diskette (DISK) function
- Storing data on cassettes or diskettes
  - Tape Cassette (TCS) function
- Switching to DOS and Executing a DOS Command
  - DOS run

- Copying a Report to a DOS File
  - SAV run
- Transferring a DOS File to the MAPPER System
  - RES run
- Executing Harvard Graphics Software
  - HG run
- Executing Lotus 1-2-3 Software
  - LOTUS run
- Executing Microsoft Word Software
  - WORD run

To save a report or range of reports on tape, contact your coordinator.

Printing MAPPER Reports

The print capabilities of the MAPPER system allow you to create hard copies of the information contained in the MAPPER reports.

These are the functions and their calls for printing reports with the MAPPER system:

- Printing reports on the system printer
    - Print (PR) function
- Printing reports to a terminal auxiliary print device
    - Auxiliary (AUX) function
- Listing auxiliary printers configured
    - Device (DEV) function

Performing Interactive MAPPER Operations

These functions allow you to pass data or messages between MAPPER sites and interact with the operating system:

- Sending reports or results to another MAPPER station
  - Send Report (SEND) function
- Communicating online with other users
  - Call (CALL) function
- Sending reports or results to another MAPPER user
  - Send Report to User (SNU) function
- Sending messages to another MAPPER station or to your own station
  - AL (Alarm) run
- Returning a message to a queue for sending at a later time
  - SNOOZE run
- Sending messages to another MAPPER station
  - Station-to-Station Message (SS) function
- Acknowledging an outstanding message
  - Acknowledge Message (OK) function
- Accepting an outstanding message
  - Message Waiting (MSG) function
- Starting a MAPPER run at another site and optionally send data
  - Remote Run (RR) function
- Executing a UNIX shell script or IBM job stream, or passing a MAPPER report to a UNIX program
  - Start function
- Starting a WFL job
  - Start function

- Executing a MAPPER run in background
    - Background Run (BR) function
- Scheduling a background run
    - SCHEDULE run
- Accessing the UNIX operating system
    - UNIX Interface (UNIX) function
- Using a MAPPER terminal in demand mode
    - Remote Symbiont Interface (RSI) function

Section 4
MAPPER Functions and Runs

You may find additional information specific to your MAPPER system by pressing **ReadMe** from the sign-on screen.

This section lists the MAPPER functions and runs in alphabetical order by the function call. For example, the S (Search) function is listed before the SV (Save Report Version) run. The first part of this section includes the following:

- Getting started with functions and runs
- List of functions and runs
- How to use this section

The remainder of this section presents the MAPPER functions and runs alphabetically by call names.

Getting Started with Functions and Runs

If you are a new MAPPER system user, you can learn the basics for using functions and runs by reading the *Manual Functions Training Guide*. Since this section presents functions and runs alphabetically by their calls, the easiest and most common functions and runs do not come first.

If you are new to the MAPPER system, familiarize yourself with the following functions and runs in the order listed:

- Sign on MAPPER Software and Sign off MAPPER Software (X)
- Display Report (D)
- Drawer Table of Contents (T)
- HELP
- Duplicate Report (XR) and Add Report (AR)
- SOE Update
- Add Line, Duplicate Line, and Delete Line
- Paint (PNT), Resume (RSM), Abort, and Release Display (^)
- Sort
- Replace Report (REP)
- Locate (LOC) and Search (S)
- Totalize (TOT)
- Print (PR) and Auxiliary (AUX)
- Add To (ADTO) and Add On (ADON)
- Search Update (SU) and Update (UPD)
- Acknowledge Message (OK), Station-to-Station Message (SS), and Send Report (SEND)

List of Functions and Runs

Table 4-1 lists the functions and runs alphabetically by name. The rest of Section 4 presents each function and run alphabetically by function call names.

Table 4-1. Alphabetical List of Functions and Runs

Function Name	Function Call
Abort	Press the **Abort** key *
Acknowledge Message	OK *
Add Line	See Table 4-2 *
Add On	ADON *
Add Report	AR *
Add To	ADTO *
Alarm	AL *
Append Line	See Table 4-2 *
APT Register	APTREG
APT Select	APTSEL
Arithmetic	A
Auxiliary	AUX
Background Run	BR
Binary Find	BF
Cabinet Switch	C or CS *
Calculate	CAL
Calculate Update	CALU *
Calendar	CALENDAR *
Call	CALL *
Change	CHG
Clock	CLOCK *
Combine	COMBINE
Communications Output Printer	COP *
Compare Data	CMP
Copy	COPY *
Copy Report to DOS	SAV *
Count	CNT
Create File	FILE
Create Result Copy	RSLT *
Create Temporary Format	VIEW *
Date	DATE
Decode Report	DECODE
Delete	DEL

continued

* Documented in the online help system only (enter **help,*call***).

Table 4-1. Alphabetical List of Functions and Runs (cont.)

Function Name	Function Call
Delete Line	See Table 4-2 *
Delete Report	DR *
Delete Share	DELSHR
Device	DEV *
Diskette	DISK *
Display Alternate Format	F*n* *
Display and Hold Headings	DH *
Display Line Count	DLC *
Display Report	D *
DOS	DOS *
Downline Load	DLL *
Drawer Password	DPW *
Drawer Table of Contents	T *
Duplicate Line	See Table 4-2 *
Duplicate Report	XR *
Element	ELT
Element Delete	ELT-
Encode Report	ENCODE
Entry	ENTRY
Exit MAPPER System	EXIT *
Extract	EXT
Find	F
Form Generation	FORMGEN
Form Generation (APT)	FORMG
Harvard Graphics	HG *
Help	HELP *
Hold Characters on Screen	HC*n* *
Hold Lines on Screen	H*n* *
Index	I
Index User	IU
Information	INFO
Insert Line	See Table 4-2 *
Iterative Binary Find	IBFN *
Iterative Calculate	ICAL
Iterative Count	ICNT *
Iterative Date	IDAT *
Iterative Find	IFND *
Iterative Sort	ISOR *
Iterative Search	ISRH *
Iterative Totalize	ITOT *
Kill	KILL
Language	LANG *
Line Control	L *

* Documented in the online help system only (enter **help,*call***).

Table 4-1. Alphabetical List of Functions and Runs (cont.)

Function Name	Function Call
Line Zero	LZ *
Lists	LISTS
Locate	LOC
Look Switch	LOOKSW *
Lotus 1-2-3	LOTUS *
Match	MA
Match Update	MAU *
Message Waiting	MSG
Microsoft Word	WORD *
Move Line	See Table 4-2 *
Name	NAME *
Names	NAMES *
Paint	PNT *
Pass	PASS
Password	PSW *
Print	PR
Pull	PULL
Put Line	See Table 4-2 *
Read Password	RPSW *
Reformat Report	RF *
Release Display	^ *
Remote Run	RR *
Remote Symbiont Interface	RSI *
Replace Report	REP *
Report and Line Limits	LIMITS *
Resume	RSM *
Retrieve File	RET
Retrieve File	RET [P]
Retrieve Report	RETR
Run Status	RS
Save Report Version	SV
Schedule	SCHEDULE *
Search	S
Search Update	SU *
Send Report	SEND
Send Report to User	SNU *
Shift Display	S*n* *
Sign off MAPPER Software	X *
Sign on MAPPER Software	
Snooze	SNOOZE *
SOE Update	*

* Documented in the online help system only (enter **help,***call*).

Table 4-1. Alphabetical List of Functions and Runs (cont.)

Function Name	Function Call
Sort	SORT
Sort and Replace Report	SORTR *
Start	START
Station-to-Station Message	SS
Stop	STOP
System	SYSTEM
Tape Cassette	TCS *
Terminal Width	TW *
Totalize	TOT
Transfer DOS File to MAPPER	RES *
Undo	UNDO *
UNIX Interface	UNIX
Update	UPD
Validate	VALIDATE
Yank Line	See Table 4-2 *

* Documented in the online help system only (enter **help,***call*).

All calls (except line edit functions) can be entered from the home position of MAPPER screens or from any control position on the MAPPER control line.

Line Edit Functions

Table 4-2 lists the function calls for line edit functions.

Table 4-2. Line Edit Functions

Function Name	Function Call
Add Line	♦]*q*+[*predfl*]
Append Line	♦][*n*]**A**[*b*]
Delete Line	♦]*q*-
Duplicate Line	♦]*x***X**[*q*]
Insert Line	♦]*x***I***/*[,*q* | -*//*]
Move Line	♦]*x***M***/*[,*q* | -*//*]
Put Line	♦]**P**[*b*]
Yank Line	♦][*n*]**Y**[*b*]

How to Use This Section

Each function and run presented in this section contains the following information, where applicable:

- **Prerequisite**. This outlines any specific conditions that must exist prior to executing the function or run.
- **Menu Procedure**. This describes how to request the function using the menu system. See "Requesting and Executing Functions" in Section 2 for the general procedure to request a function through the menu path.
- **Control Line Procedure**. This outlines the command formats and format fields required to request a function from the control line. See "Requesting and Executing Functions" in Section 2 for the general procedure to request a function from the control line.
- **Options**. Where applicable, all options available for functions and runs are described briefly in the Options table. When a function or run has a number of options that cannot be described fully in the table, guidelines or examples are provided.
- **Parameters**. Where applicable, all parameters available for functions and runs are described briefly in the Parameters table.
- **Outcome**. This describes the outcome of the function and, where applicable, describes conditions that change the outcome of a function.
- **Guidelines**. This explains any extra information you need to know before using the function and gives more insight as to how you can use the function.
- **Examples**. In most cases, examples are provided to help clarify the use and structure of MAPPER functions and runs. In some of the examples, horizontal and vertical ellipses are used to indicate that portions of the screen are not shown.

  Note: *Before trying examples that update the demonstration database, duplicate the report. This way, you will not alter or delete part of the database for others who use it after you.*

A (Arithmetic)

Use the Arithmetic (A) function to do computations using either a feature called the MAPPER calculator or equations you define in a report.

For example, if you want to do some calculations on a set of values that are not in a report, you can call the A function and use it as you would a pocket calculator.

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Type your arithmetic expression or formula after the SOE character and transmit.

   For information on creating arithmetic expressions and using predefined equations, see other subsections in this section.

Menu Path

Select Task
    Perform Math Operations
        Use the Arithmetic Function

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

A

Displays the MAPPER calculator.

A *rd*[*c*]

The calculator becomes active and displays the specified report. See "Using Predefined Equations" in this section for more information.

Format Fields

rdc Report containing the predefined expressions. (Type **a** - to process the current report.)

Outcome

The A function does the calculations and displays the answers on the screen in a drawer A result.

Guidelines

Toggling Between Result and Calculator

To display the result when the calculator is on display, press **Contrl**. To display the calculator again, when the result is displayed, press **Resume**.

Entering Equations

You can enter equations with or without blanks surrounding the mathematical operators; however, when using the Arithmetic (ART) run statement, you cannot use spaces between operators. See the *Run Design Reference* for more details.

Operators, Priorities, and Functions

For a list of operators and their priorities, and for a list of available mathematical and trigonometric functions, see Appendix E.

Creating a Report for Predefined Equations

Follow this procedure to create a report for predefined equations:

1. Add a report to a freeform drawer such as drawer A, using the Add Report (AR) function.
2. If a heading divider line does not already exist, insert one directly below the headings.
3. Type values and equations as described below:
   - Begin the first value or equation on the second line of report or in the second column of the first line so the SOE character in the arithmetic calculator screen does not cover the first character in the equation.

- End all values and equations, except the last one, with a semicolon.
- Precede variables with a tab character. Then, when you use the A function later to process the report, you can easily tab to variable positions and type new values.

Using Predefined Equations

To use equations already defined in a report, follow this procedure:

1. Start the A function using the control line procedure, specifying the report containing the predefined equations. (Press **Format** to read about the control line procedure.)
2. Do you need to change the values of variables in the predefined equations?

   Yes: Tab to the variables and replace them with the values to be used in your calculations. Move the cursor to the line below the last expression and transmit.

   No: Move the cursor to the line below the last expression and transmit.

How Variable Names Are Assigned in the Arithmetic Function

You can choose between assigning values to variable names and allowing the system to assign them.

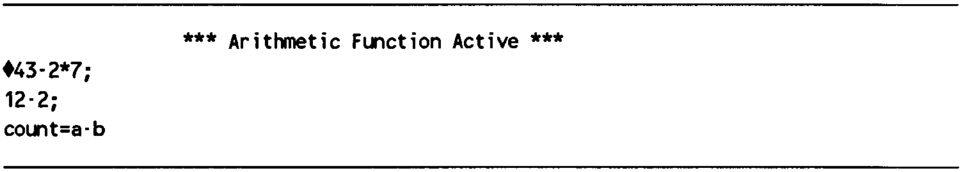
Assigning names: You can assign a value to a certain variable name. For example, the expression INPUT=35.6 assigns the value 35.6 to the name INPUT. The expression DATA=4+5 assigns the value 4+5 to the name DATA.

Allowing the system to assign names: If you do not assign a value to a certain name, the A function does so for you. If you type the expression 45-7 with no assigned name, the function labels it with a single letter. The function names the first unassigned value A, the second B, the third C, and so on.

See the example "Entering Equations and Variables."

Example: Entering Equations and Variables

The following screen example shows two equations without user-assigned variable names and one equation with a user-assigned name (count).



```
                    *** Arithmetic Function Active ***
♦43-2*7;
12-2;
count=a-b
```

The result displays three final values:

- A = 29. The system gives the first set of computations (43-2*7) the variable name A.
- B = 10. The system gives the second set of computations (12-2) the variable name B.
- COUNT = 19. The system displays the last value of the assigned variable name COUNT.

APTREG

Use the APTREG run to register fields from an existing report or drawer into the data dictionary.

Note: *The APTREG run is part of the Application Power Tools (APT) feature of MAPPER software. For more information about the APTREG run, the data dictionary, and other APT features, see Appendix F.*

Prerequisite

The report or drawer must exist on the system and cannot contain any passwords.

How to Access the Run

Menu Access Procedure

Procedure	1. Type **apt** on the control line and transmit. 2. Select **APTREG** from the APT menu. 3. Fill in the run form. For help while filling in the form, press the **Help** key. 4. If you want to register more reports or drawers, fill in the form again. When finished, press **Resume**.

Control Line Procedure

Procedure	Enter the following format on the control line.
Format	**APTREG** Displays the run form. For help while filling in the form, press the **Help** key.

Outcome

The APTREG run enters the field names and sizes from the specified reports into the data dictionary. If you register fields from an entire drawer or from report 0, the APTREG run also registers the edit codes in the data dictionary.

If any fields already exist in the data dictionary, the APTREG run displays a message and does not register those fields.

Guidelines

When filling in the run form, you can press **List** to display a menu of reports registered in the data dictionary for the current application. Select the reports to process by typing an X in the **Enter** field.

After the fields are registered, you can use the CHG function of the ENTRY run to add documentation for each field.

For More Information

For more information about the APTREG run, enter `apt` on the control line, tab to **APTREG**, and press **Help**. For more information about the data dictionary and the runs available with Application Power Tools (APT) feature of MAPPER software, see Appendix F.

APTSEL

Use the APTSEL run to select a default application name that appears in the APT run forms.

Note: The APTSEL run is part of the Application Power Tools (APT) feature of MAPPER software. For more information about the APT features, see Appendix F.

Prerequisite

You can select only those applications controlled by your department data dictionary.

How to Access the Run

Menu Access Procedure

Procedure

1. Type **apt** on the control line and transmit.
2. Select **APTSEL** from the APT menu.
3. Fill in the run form. For help while filling in the form, press the **Help** key.

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

APTSEL
Displays the run form. For help while filling in the form, press the **Help** key.

APTSEL{,*appnm*|,c}
Changes the default application name to the name you specify, or clears the default application name.

Format Fields

appnm	The name of the application to use as the default.
c	The letter c to clear the default application.

Outcome

The APTSEL run changes the default application to the name you specify, or clears the default application if you specified no application name. The default application name appears in the APT run forms.

For More Information

For more information about the APTSEL run, enter `apt` on the control line, tab to **APTSEL**, and press **Help**. For more information about the data dictionary and the runs available with Application Power Tools (APT) feature of MAPPER software, see Appendix F.

AUX (Auxiliary)

This section does not apply to the BTOS II MAPPER System. Use either the Print (PR) function or the COP (Communications Output Printer) run.

Use the Auxiliary (AUX) function to print reports on an auxiliary printer connected to a terminal.

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Fill in the function form. For help while filling in the form, press **Help**.

Menu Path

Select Task
    Print
        Print on an Auxiliary Printer

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

AUX
Displays the function form.

AUX *sn*[*sl rdc f*]
Queues the report to the auxiliary printer on the specified station.
1100: This format is not available on OS 1100 MAPPER Systems.

Format Fields

sn Station number of the terminal to which the printer is attached.

sl Remote site letter for printing at another site. (For a list of sites, press **Remote** from the active screen.) Default = local site.

f Format of the report to print: 0 through 25. Default = basic, unshifted format.

Outcome

Queues the report to be printed.

Note: *For printing to start, the sign-on screen must be displayed on the station you specify in your request.*

Suspending or Stopping Printing

Suspending printing: To suspend printing temporarily, press **Abort**. The printing stops and the sign-on screen is displayed. You can then sign on to the system. Printing resumes the next time you sign off the system.

Stopping printing: To stop printing, follow this procedure:

1. Are you at the station you specified in the AUX **Printing** request?
   Yes: Suspend printing by pressing **Abort**, then sign on.
   No: Sign on, if you are not already signed on.

2. Type the SX directive on the control line and transmit:

   ```
   SX [sn]
   ```

   where:

   sn Station number of the printer, if different from your station.

Outcome: Removes the report that was printing from the print queue.

Restarting the Printing

If Suspended with the Abort Key

If you pressed **Abort** to suspend the printing temporarily and then signed on, printing resumes when you sign off the system.

If Suspended by Device Error

If printing was suspended because of a device error, such as being out of paper, type the SQ directive on the control line and transmit:

```
SQ [sn]
```

where:

sn Station number of the printer, if different from your station.

Outcome: Places the report that was temporarily suspended into the print queue.

Starting the Printer at Another Station

This section on starting the printer at another station applies only to the OS 1100 MAPPER System.

Use the SI directive to start printing at another station, generally a screen bypass terminal. Use the following format:

```
SI [sn]
```

where:

sn Station number of the other station, normally a screen bypass terminal.

Outcome: Places the report into the print queue.

Skipping Pages

This section on skipping pages applies only to the OS 1100 MAPPER System.

To skip pages, follow this procedure:

1. Are you at the station you specified in the AUX request?
   Yes: Suspend printing by pressing **Abort**, then sign on.
   No: Sign on, if you are not already signed on.
2. Type the SR directive on the control line and transmit:

   ```
   SR,[sn],page
   ```

   where:

   sn Station number of the printer, if different from your station.

   page Page number relative to the page currently being printed, for example, an entry of 5 indicates printing should resume at the current page number plus 5.

   Outcome: Skips to the specified page relative to the current page being printed and resumes printing.

Example

The following SR directive resumes printing at the current page minus 4 on the printer connected to your station:

```
sr,,-4
```

Controlling the Printer

Control the printer by typing device control commands into the report to print. Remember these guidelines when typing the commands:

- Insert the commands anywhere on the lines in which you want the commands to take effect.
- Make sure the commands are the first nonblank characters on the line.
- The commands can be in uppercase or lowercase letters.

Use the following device control commands to control printing on auxiliary printers:

Command	Description
.EJECT	Advances the paper to the top of the next page before continuing to print. The period can be any character except blank. Note: This command must begin in column 1. **1100:** On OS 1100 MAPPER Systems, the command does not have to begin in column 1.
HOM	Advances the paper (same as .EJECT, but does not have to begin in column 1).
KIP *n*	Skips *n* lines on this page. If *n* is greater than the number of lines remaining on the page, the printer advances to the top of the next page.
LPP *n*	Prints *n* lines per page. Default = 66, including header and footer margins. **1100:** On OS 1100 MAPPER Systems, the default = 60.
1100: TAB *n*	Moves *n* vertical tab positions and resumes printing. This command is for model 0786 printers only. The printer handler loses the line count following the TAB command. To reset the counter, enter **hom** before the end of the page. The handler automatically executes a HOM command at the end of the report.

Emphasizing Text When Printing Reports

Insert the following control characters surrounding text you want to emphasize. The AUX function interprets and removes the control characters from the report before it prints.

Use the uppercase letters to start the operations described below and the lowercase letter to end them:

Control Characters	Purpose
~B and ~b	Specifies bold text.
~U and ~u	Underlines text.
~V or ~v	Creates a vertical line. ~V creates a vertical line and a space (|). ~v creates a space and a vertical line (|).
~X and ~x	Strikes through a character.
~^	Toggles the interpretation of device codes sent directly to the printer. See "Redefining the Control Characters" in this section. **1100:** This control character is not available on OS 1100 MAPPER Systems.

If you leave a space in the **CC** field in the function form, the auxiliary printer is not initialized, and all special control characters are printed as they appear in the report.

If the auxiliary printer cannot perform a special print function (for example, cannot do boldface), the control characters are eliminated from the printed text.

Redefining the Control Characters

The tilde (~) is the default primary control character. You can change the primary control character to any other printable character, except a caret (^).

To redefine a primary control character, enter that character in the **CC** field of the AUX function form. For example, if you have defined boldface with a **$B** and **$b** control code, enter a **$** in the **CC** field of the AUX function form.

Additionally, you can send device control codes directly to the auxiliary printer. The device code interpretation toggles on and off using a ~^ print control. For example, a ~^ followed by ^J^M directs the printer to perform a line feed (^J) and carriage return (^M). Another ~^ turns off the device code interpretation and any device code is printed literally.

1100: The ~^ control character is not available on OS 1100 MAPPER Systems.

BF (Binary Find)

Use the Binary Find (BF) function to find character strings quickly in one or more large reports or a drawer or to count occurrences of character strings.

Prerequisite

Each field or partial field being searched must be sorted in ascending order.

How to Access the Function

Menu Procedure

Procedure	1. Select menu items using the menu path below. 2. Was a report on display? Yes: Fill in the function mask. No: Fill in the function form then the function mask.
Menu Path	Select Task Find Data Perform a Binary Find

Control Line Procedure

Procedure Enter one of the following formats on the control line.

Formats

BF

Displays the function form.

BF {*rd*[*c*]|*d*[*c*]} [*f*]

Displays the function mask.

Format Fields

rdc Report to process. (Or type **bf -** to indicate the current report.)

1100: On OS 1100 MAPPER Systems, you do not need to type the minus (-) to indicate the current report.

dc Drawer to process. Use this field to search all reports in a given drawer.

f Report format in which to search for data (lets you look for data in fields beyond column 80 if you do not already have those columns on display).

Options

A Processes all line types.

B Builds an index containing the first data line of each report. Use this index with subsequent BF operations to speed up processing. See "Using an Index with the Binary Find Function" in this subsection for details.

C(S) Distinguishes between uppercase and lowercase letters.

1100: C(x) Alters the find process based on the character set order. Ordinarily the system processes the report based on the character set of the drawer. The C option allows you to choose the character set type on which to base the find. Use one of the following:

C(F)	Full character set (FCS)
C(L)	Limited character set (LCS)
C(S)	Strict comparison; distinguishes between uppercase and lowercase letters

continued

Options (cont.)

E

Finds the last occurrence of the character string instead of the first occurrence. Do not use this option with the O option.

F*n*,*n*,...

Specifies the order of sorted fields when searching multiple fields. For *n*, type the corresponding sort parameter you used for the fields from left to right. See the example "Changing the Order of Fields" for more details.

I[*n*]

Uses the index located in report *n*. Default = 2. The BF function searches the index to determine where to find the character string, assuming that the reports to be searched follow report n. See the section "Using an Index with the Binary Find Function" for details.

K

Verifies that reports are sorted in ascending order. The function displays the report at the point where it detects the sort order discrepancy and displays a message indicating whether the data is sorted.

Use with the K parameter: type **k** in the field to verify.

Note that if a report containing only blank and heading lines is included in the report range, the blank line will not verify as a valid sort order.

Because this option verifies each line of every report specified, use it with discretion on large databases.

N

Creates a result containing the last occurrence of each different character string found in the specified field and the number of occurrences of each string.

Use with the K and = or =N parameters: type **k** in the field to check. Type **=** in the field to hold the number of occurrences. Or type **=N** in any fields containing integers where you want the result to produce a subtotal. Do not use the = and =N parameters together.

Do not use this option with the O option.

O

Creates a result containing each line on which the specified character string occurs, including trailer lines. Do not use this option with the N or the E options.

continued

Options (cont.)

P — Includes all types of lines following the last find made. Use only with the N option.

Q — Finds a string that occurs only once in the report for a faster search. When used with the O option, no trailer lines are included.

R*x*-*y* — Searches reports in a range from report *x* to report *y*. All reports in the range must exist. See the Guidelines subsection for details.

S — Searches each report separately. With this option, a report field can be scanned over an entire drawer even if that field is not sorted over the entire drawer. The scan stops after finding the first item and continues with the next report only when the user resumes.

To create a result that shows finds from all reports scanned, use the O option with the S option.

U — Sets an update lock on the report in which a find is made or would have been made. Use this option to find the correct report in a range of reports for entering new data.

@ — Searches for blank lines or fields at the end of a report. Use this option to find a place to enter data. Use with the K parameter: type **k** in the fields where you are looking for spaces. Do not use this option with the R option.

/ — Searches for a slant as data when used with the / parameter: type / in the columns where you are looking for a slant.

Parameters

The character string you want to find is the most common parameter to use with the BF function. You can also specify the following parameters:

/ Searches for a slant (/) character when used with the / option. Without the / option, a slant appearing in a parameter line shortens that field and a partial field is processed.

= Specifies the field in which to place the total number of occurrences of each string when used with the B or N option. Do not use this with the =N parameter.

=N Specifies the field in which to place a subtotal. Do not use this with the = parameter.

K Specifies the fields to process when used with the B, K, N, or @ option.

R Specifies a search for a range of character strings (for example, blackbox1 through blackbox4). You can perform a range search on only one field.

Outcome

- The function displays the report at the line where the data is located.
- If the data is not found, a system message is displayed.
- Since the function does not consider blank lines a sort order discrepancy, it ignores them.

Guidelines

Searching for More Than One Character String

To search for multiple strings in different fields, enter the character strings under different headings on the same line. Before using the BF function, sort each field in ascending order. Either sort the leftmost field first, the next field to the right second, and so on, or use the F option to specify the order of the sorted fields.

Searching for a Range of Strings

To search for a range of character strings, follow this example:

1. Type the characters for the lower end of the range in the first line under the mask in the correct field.
2. Type r in the first character position in the next line.
3. In the same line as the R, type the characters for the higher end of the range in the same field as the lower end. Then transmit.

Searching a Range of Reports

When searching a range of reports, make sure that none of the reports is empty (containing only heading lines). If an empty report is encountered and a find is made in a subsequent report, the BF function does not search any reports preceding the empty report.

When searching an entire drawer, the BF function begins with report 3. (To process reports 1 and 2, use the R option.)

Sort the fields to be searched in all of the reports across the entire range of reports. For example, if the specified field in report 10 contains items starting with the letter A and continues through letter C, report 11 could continue with letters D through H, report 12 with letters I through M, and so on.

IBFN Run

1100: On OS 1100 MAPPER Systems, you can use the IBFN run to generate BF function masks that you can save, alter, and use again. For information on how to use the iterative runs, see Section 5.

Using an Index with the Binary Find Function

When to Use an Index

When you have many reports to search, use an index to accelerate the find process on subsequent searches. The B option builds the index containing the first data line of each report searched. To use this index, specify the I option.

Using Empty Reports

If any reports in the range contain only headings, add one blank line in each of these reports before building the index. After the index is built, delete the blank line.

1100: On OS 1100 MAPPER Systems, empty reports may exist within the range of reports being processed.

Creating and Saving the Index

Use the K parameter to specify the target field or fields. Be sure the same target does not appear in more than one report.

1100: On OS 1100 MAPPER Systems, the target may appear in adjacent reports.

You must replace the index result into the report that immediately precedes the first report in the range. Report 2 (the default) is used as the index report unless you specify another report with the I option or a range of reports with the R option.

Examples

Example 1: Counting the Number of Strings in a Field

```
n
*St.Order . Product .Ord.Cust.
*Cd.Number.  Type   .Qty.Code.
*==.======.=========.===.====.
 ** ****** ********* *** ****
                     =   k
```

n Perform a count operation.

k Count the number of unique strings in the Cust Code field.

= Place the result of the count operation in the Ord Qty field.

Example 2: Changing the Order of Fields

Assume your database was sorted using sort parameters in the following sort function mask (note that the last character of the Product Type field is not being sorted):

```
*St.Order . Product .Ord.Cust.
*Cd.Number.  Type   .Qty.Code.
*==.======.=========.===.====.
 ** ****** ********  *** ****
    2      1             3
```

To process this sorted database with the BF function, use the F option as shown in this Binary Find function mask. Note that the last character of the Product Type field is not being searched.

```
f2,1,3
*St.Order . Product .Ord.Cust.
*Cd.Number.  Type   .Qty.Code.
*==.======.=========.===.====.
 ** ****** ********* *** ****
    123445 blackbox/     arco
```

f2,1,3 Specify the sort hierarchy of the fields.

BR (Background Run)

Use the Background Run (BR) function to start a MAPPER run and execute it as a background process, freeing your terminal for other processing.

Prerequisites

The background run must not contain the following run statements, because they all attempt to access a terminal screen, and there is no terminal associated with backgound runs: DSG, DSM, DSP, DSX, GOC (when display requested), GS (when display is requested), HST, ITV, OTV, OUM, OUT, OUV, RDB, REL, RSI, SC, XIT, XUN.

Note: *Since this list encompasses several different kinds of MAPPER systems, some run statements may not be available on your system.*

1100: On OS 1100 MAPPER Systems, you can use the DSG, OUT, and SC statements if you send the output to another terminal. You can also use the RDB statement in background runs.

In addition to the statements listed above, the background run must not contain a BR statement. In other words, a background run may not start another background run.

1100: On OS 1100 MAPPER Systems, have the coordinator register the run in one of the following ways:

- As a normal run eligible for background execution (you can execute the run either normally or in background)
- As a background run only

How to Access the Function

Menu Procedure

Procedure	Select menu items using the menu path below.
Menu Path	Select Task Use Utilities Start a Run Start a Background Run

Control Line Procedure

Procedure Enter the following format on the control line.

Format **BR** ***run*****[,*****d1*****,*****d2*****...]**

Starts a run in background.

1100: In addition to the BR call, you can use the BRX call to eliminate the completion message:

BRX ***run*****[,*****d1*****,*****d2*****...]**

Starts a run in background and does not send a completion message.

If the run is registered as background only, you can omit the BR call and just enter the run name.

Format Fields

run Name of the MAPPER run for background processing.

d1,d2 Data passed to the background run, delimited by commas. You can pass a maximum of 40 data items.

1100: On OS 1100 MAPPER Systems, you can pass a maximum of 39 data items.

Outcome

- The run starts in background, and the active screen is displayed.
- If a report was on display when you started the run, the report redisplays after the run executed.
- If the background run fails, you receive a message indicating the error. See the Message Waiting (MSG) or Acknowledge Message (OK) functions for information on receiving messages.
- **1100:** When the background run finishes, you receive a completion message. To request that the system not send a completion message, use the BRX call instead of the BR call.

Guidelines

Maximum Number of Runs

The maximum number of background runs that can be active in the system is set by the MAPPER system coordinator. If the maximum is already active, the system queues all subsequent background runs.

1100: On OS 1100 MAPPER Systems, the maximum number of background runs that can be active in the system is set by the system operator or the coordinator. The system does not queue background runs if the maximum is already active; instead, a system message is displayed.

Checking Run Status

To check the status of the run, use the System function.

1100: On OS 1100 MAPPER Systems, use the Run Status (RS) function to check the status of the run.

Terminating a Background Run

To terminate an active background run, use the Kill function. To terminate a queued background run, contact the MAPPER system coordinator.

▽ **1100:** On OS 1100 MAPPER Systems, use the Stop function to terminate a background run.

CAL (Calculate)

Use the Calculate (CAL) function to compute, compare, and replace numeric data, character strings, dates, and times in a report. You can use the CAL function in many situations including the following:

- If you need to perform certain operations only when specific conditions are met, use conditional statements in the CAL function.
- Since you can perform several operations at once in CAL equations, use the CAL function in place of multiple Totalize (TOT) or Date functions.
- If you need to manipulate character strings or perform complex computations in reports, use the CAL function in place of extensive loops and logic statements in runs.

The following tasks are just a few of the many operations you can perform with the CAL function:

- Copying or moving data between fields when a certain condition is met
- Comparing two dates and performing calculations whenever one date is later than the other
- Processing sliding pay rates or discount rates based on factors such as number of hours or days worked or total dollar amount of an order
- In a scheduling database, tracking the number of minutes, hours, or days ahead of or behind schedule and keeping a running subtotal and average by product type
- Calculating the number of workdays left to meet a deadline
- Looking for a character string and replacing it with another
- Looking for duplicated information in a report and removing it
- Obtaining the highest or lowest number across several fields or within each field
- Creating a result containing only those lines that meet a certain condition

Use your imagination. Generally, if you have a complex problem of almost any kind, or one that involves multiple equations, even if not too complex, you can use the CAL function to solve it.

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Was a report on display?

   Yes: Fill in the function mask.

   No: Fill in the function form then the function mask.

Menu Path

Select Task
    Perform Math Operations
        Calculate Data

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

CAL

Was a report on display?

Yes: Displays the function mask.

No: Displays the function form.

CAL *rd*[*c f*]

Displays the function mask.

Format Fields

rdc Report to process.

f Report format in which to process data (lets you process data in fields beyond column 80 if those columns are not already displayed).

Options

A Processes all line types.

C Conditionally includes lines in the result: after all equations are processed, only those lines that meet a true condition based on the last IF statement are included in the result. See "Computing Using Conditional Statements" in this section.

E Fills result fields with spaces whenever the result is 0.

I Produces integer results, dropping any fractional part from the result.

J(x) Justifies numeric result values in fields to *x*; *x* is one of the values listed below. When displayed, leading zeros appear to the left of a value, nonsignificant zeros to the right. Neither affect the value.

- c Eliminates leading and nonsignificant zeros. Inserts commas every three digits in the integer portion of the result. Right-justifies values.
- l Eliminates leading and nonsignificant zeros. Left-justifies values.
- r Eliminates leading and nonsignificant zeros. Right-justifies values.
- x Eliminates leading zeros. Left-justifies values, and fills fields with nonsignificant zeros.
- z Eliminates leading and nonsignificant zeros. Right-justifies values, and fills leading spaces with zeros.

K*n* Initializes value labels to *n* on the first pass. Default = 0. See "Calculate Function Labels" in this section.

L Lists all value labels and their final values at the end of the report.

N*n* Substitutes a numeric value *n* for nonnumeric fields. Default = 0. Nonnumeric fields are fields containing either nonnumeric data in the leftmost column of the field or no data.

O Omits all data lines from the result. Includes only heading lines and value label names with their final value in the result.

continued

Options (cont.)

R*n* Rounds numbers to the nearest *n*. Use with the E option to round entry counts. The range for *n* is .000000000000001 to 100000.

For example, r.001, rounds the values to the nearest thousandth. The option r10000, rounds the values to the nearest ten thousand.

S(S) Distinguishes between uppercase and lowercase characters when processing character strings.

▽ **1100:** Use the S(x) option instead of the S(S) option.

S(*x*) Defines character set interpretation as *x* (F, L, or S). Note that the system compares limited character set (LCS) strings to LCS internal codes, full character set (FCS) strings to FCS internal codes, and interprets uppercase and lowercase letters the same.

F Use FCS internal codes. Use only when processing LCS reports.

L Use LCS internal codes. Use only when processing FCS or full character set upper (FCSU) reports.

S Use strict character set internal codes to differentiate between uppercase and lowercase characters. Only with FCS report.

T Includes both processed and unprocessed lines in the result. Do not use the T option if you want to include only the type of line being processed.

V Processes only those equations whose result values are calculated from valid data. Equations containing invalid data are not included in the result, and the receiving label is not altered. See "Formulating Equations with the Calculate Function" in this section.

Invalid data includes field labels representing nonnumeric data and value labels whose value was calculated from nonnumeric data or an invalid date. Nonnumeric fields are fields containing either nonnumeric data in the leftmost column of the field or no data.

continued

Options (cont.)

X Excludes invalid values in minimum, maximum, sum, and average computations (MIN, MAX, SUM, AVG, VMIN, VMAX, VSUM, VAVG) and in operations specified by vertical operators. See "Computing Vertically Using the Calculate Function" in this section and Appendix E.

* Flags all invalid results with an asterisk following the value.

Invalid data includes field labels representing nonnumeric data and value labels whose value was calculated from nonnumeric data or an invalid date. Nonnumeric fields are fields containing either nonnumeric data in the leftmost column of the field or no data.

Parameters

Field Labels See "Calculate Function Labels" in this section for information on field labels.

Vertical Operators See "Computing Vertically Using the Calculate Function" for information on vertical operators.

Equations See "Formulating Equations with the Calculate Function" in this section for information about equations.

Outcome

- The function processes the equations and creates a result.
- You can place values produced by equations into report fields or into labeled storage areas for later use. The maximum size of the fields or labels is 18 characters.

  The resulting values overwrite data previously in the receiving field or label.

- The function either rounds or expands resulting values that are not integers to fill the size of the field.
- The function right-justifies integers if possible.
- The function performs all of the equations in the order you type them for each line of data.
- If you specify a receiving label that is not large enough to contain the output format or if the result value of the expression is not a valid date or time, the fields are filled with asterisks and are assigned a value of zero.
- See the examples, guidelines, and other subsections in this section for outcomes specific to the CAL operation you perform.

Guidelines

Related Functions

For simple calculations on reports, use the TOT and Date functions. For calculations outside of reports, use the Arithmetic (A) function.

ICAL Run

You can use the ICAL run to test a set of calculations, change them and try again, or save them for later use. Calculations using the ICAL run operate exactly as calculations using the CAL function. See the online help system (HELP,ICAL) for more information.

Extracting Data

To create an update result that allows you to blend the changed lines back into the original report or delete lines from the original report, use the Calculate Update (CALU) function. See the online help system (HELP,CALU) for more information.

Calculate Function Labels

A label is a name for a data value in an arithmetic or relational expression. You can use three types of labels: field, value, and constant.

Field Labels

A field label is a single alphabetic character that identifies fields to process. When you enter a letter in a field in the function mask this letter becomes the label identifying that field. You can enter field labels anywhere in the field and in any order.

Whenever you specify unique field labels for each field, they are considered single field labels and, since they represent single values, can be used as operands.

A multiple field label is a field label that identifies more than one report field. You cannot use multiple field labels as operands because they do not represent single values. See "Formulating Equations with the Calculate Function" in this section for more information.

Value Labels

A value label is a one- to six-character name that identifies a single numeric value. Use value labels in place of actual values or expressions to simplify typing equations. You can define your own value labels or use constant labels set by the system. Once defined, the value does not change until reset by another equation.

To define a value label, use the following format:

label=value or *label=expression* (for example, **abc=a+b+c**)

The label must begin with an alphabetic character and can contain only alphanumeric characters and special characters $, %, !, and ?.

Internally, value labels contain 18 characters and begin with a value of 0. You can use the FIRST conditional statement to initialize value labels to another value, such as spaces. See "Computing Using Conditional Statements" in this section.

Note: *Label names you define must not be the same as constant labels or internal functions. See Appendix E for a list of internal functions.*

Constant Labels

Constant labels are value labels predefined within MAPPER software. You cannot store characters in constant labels or use them as receiving labels (except for the LT label).

LINE Line number of data currently being processed (does not count heading lines and counts only those lines of the type of line you specified for processing). The LINE label is set to 1 initially and increases by 1 each time a new data line is processed.

LT Line type designator of the data line currently being processed (for example, a tab or *).

PI Value of pi (3.14159265358979).

RLINE Report line number currently being processed (counts heading lines and all line types, regardless of the type you are processing).

Formulating Equations with the Calculate Function

In the CAL function, an equation is an operation that stores the result of an expression in a value label or a field in a report represented by a field label.

In an equation, always place the field label or value label to the left of the equal sign and place the expression to the right of the equal sign.

Entering Equations

Enter equations in the second parameter line of a manual function mask. In a function mask, you can let the equation wrap to the next line if it is longer than the width of the screen.

You can process fields or labels containing up to 18 characters.

Place a semicolon (;) between each equation.

You can use any number of spaces or tab characters within equations (for example, `a = b + c`); however, in the CAL run statement, you must either remove the extra spaces or place apostrophes before and after the equation.

Equation Format

Following is the format to use in CAL equations:

```
receiving-label[,options]=expression[; (more equations) ]
```

where:

receiving-label Field label or value label to hold the result of the expression.

options Equation options E, I, J(x), Kn, Nn, Rn, V, X, and *. Use to specify options on individual equations. To remove a function option from an equation, enter a minus sign (-) after the equation option. For example, J-, K-, V-.

expression Any number of labels, values, and operators that combine to produce a single arithmetic or logical value. See "Arithmetic Expressions" and "Relational Expressions" in this section for more information.

Using Field Labels as Receiving Labels

If you use a field label as a receiving label, the result overwrites the original value of the field label.

If you use a multiple field label as a receiving label, the CAL function repeats the equation once for each field identified.

Using LT Constant Label as Receiving Label

Since you cannot store characters in constant labels, you cannot use them as receiving labels, with one exception: you can store a character into the constant label LT. This changes the line type of data currently being processed.

Partial Receiving Labels

Use the following format for partial receiving labels:

label(*x*-*y*)

where:

x Starting character within the label.

y Number of characters.

Arithmetic Expressions

Arithmetic expressions consist of labels, values, and arithmetic operators that combine to produce a single arithmetic value.

For example, this expression adds the value of *x* to the value of *y* and multiplies the sum by 44:

(*x*+*y*)*44

Arithmetic operators are symbols you use to specify arithmetic operations. You must specify, not imply, the operators in CAL equations. For example, to multiply *x* by *y*, use *x***y* rather than *xy* or (*x*)(*y*).

See Appendix E for information about valid operators.

Relational Expressions

Relational expressions consist of labels, values, and relational operators and are used to compare values or character strings. See also "Computing Using Conditional Statements" in this section. A relational expression also can be used as the value of 1 if the comparison is true and the value of 0 if the comparison is false. See the example "Computing the Difference in Workdays, Method 3."

Table 4-3 shows the priority of relational and logical operators.

Table 4-3. Priority of Relational and Logical Operators

Priority	Operator	Description
First	=	Equal to
	<> or ><	Not equal to
	>	Greater than
	<= or =<	Not greater than (less than or equal to)
	<	Less than
	>= or =>	Not less than (greater than or equal to)
Second	&	AND (logical operator)
Third	,	OR (logical operator)

Logical operators are symbols you use to perform logical, not numeric, comparisons. Each statement compared is either true or false itself, and the result of a logical operation is true or false.

The following tables show the results of all combinations of AND and OR operations.

AND Operation

This is the format for an AND operation:

```
Statement1 & Statement2
```

This shows the possible results of all combinations of an AND operation:

Statement 1	Statement 2	Result
False	False	False
True	False	False
False	True	False
True	True	True

OR Operation

This is the format for an OR operation:

```
Statement1 , Statement2
```

This shows the possible results of all combinations of an OR operation:

Statement 1	Statement 2	Result
False	False	False
True	False	True
False	True	True
True	True	True

Built-in Functions

A built-in function is a predefined mathematical or other kind of operation you can perform against values and any kind of label except a multiple field label (unless it is the same as the receiving label).

See Appendix E for a list of valid mathematical functions. See also Table 4-4 for valid X,Y parameters when using built-in functions in CAL equations.

To use a built-in function, set a value label or field label to equal the value produced by the function. In the following example, the value label sqroot holds the square root of the result of the specified expression:

```
sqroot=sqrt(label1*label2/label3)
```

Table 4-4 lists valid x and x,y entries depending on the kind of format required for a particular built-in function (FUNCTION equals the built-in function call such as SQRT or VSUM):

Table 4-4. Format Type and Valid x or x,y Entries for Built-in Functions

Format Type	Valid x or x,y Entries
FUNCTION(*x*)	Value, arithmetic expression, or any kind of label except a multiple field label (unless it is the same as the receiving label).
FUNCTION(*x,y*)	Values or any kind of labels except multiple field labels (unless it is the same as the receiving label).
FUNCTION(*x1,...xn*)	Values or any kind of labels. If you use a multiple field label, the function acts on all fields identified by multiple field labels.

Determining the Kind of Data in a Report Field (DEF)

Use the Definition (DEF) built-in function to determine what kind of data a field contains.

This is the format of the DEF function:

```
DEF(field-label)
```

The DEF function produces one of the following values:

Value	Contents of Field
0	All spaces or tab characters or both
1	All numeric characters
2	All alphabetic characters
3	Alphabetic and numeric characters
4	All special characters
5	Special and numeric characters
6	Special and alphabetic characters
7	Special, alphabetic, and numeric characters

The contents of the field may include the characters indicated or the characters and spaces.

Computing Horizontally Using the Calculate Function

With the CAL function, you can perform computations in fields on each line and store the resulting values in a field on that same line. Follow this procedure:

1. Type field labels (single alphabetic letters) in the fields you want to process and in the field that will hold the result.
2. Type an equation using the field labels and arithmetic and relational operators.

See the example "Computing Horizontally" in this section.

Computing Vertically Using the Calculate Function

Vertical operations produce values that are associated with both the current line being processed and other lines in the report.

Follow this procedure to compute vertically:

1. Type field labels (single alphabetic letters) or vertical operators, or both, in the fields you want to process.
2. Type an equation using the field labels and arithmetic and relational operators.

Vertical Operators

Vertical operators are symbols you use to specify an arithmetic operation to be performed on all the values in a field. Place one of the following operators in a field either in place of a field label or in addition to a field label:

+	Sum of all values in the field
/	Average of all values in the field
<	Lowest value in the field
>	Highest value in the field

Outcome Using Vertical Operators

For each data line, the function processes all of the equations you specify before it processes the vertical operators. It places resulting values at the end of the result under the appropriate fields.

See the example "Totaling Fields Vertically."

Computing Using Conditional Statements

Conditional statements test conditions to determine whether or not the next equation is processed, and they specify an alternative equation to process if a condition is not met. Table 4-5 lists and describes the formats for conditional operators.

Table 4-5. Formats and Description of Conditional Operators

Format	Description
IF:*expression;*	Tests the condition using a relational expression.
THEN:*equation;*	Processes the equation when the condition is met.
ELSE:*equation;*	Processes the equation when the condition is not met.
FIRST:*equation;*	Processes the equation only the first time it is encountered.
SKIP:*expression;*	Skips remaining equations and continues on next data line.
EXIT:*expression;*	Stops processing equations and displays a result.

Using SKIP and EXIT

Use the SKIP and EXIT statements to increase efficiency of the CAL function. For the ***expression*** field in the format, you normally use a relational expression to specify the condition when equations should be skipped. See the example "Conditionally Skipping Equations."

SKIP outcome: When the SKIP expression is true, processing on the current data line stops immediately. You might have several equations left to process for that data line, but the SKIP statement indicates they do not need to be processed whenever the expression is true. Processing continues on the next data line starting with the first equation.

EXIT outcome: The EXIT statement works like the SKIP statement, except that when the expression is true, all CAL equations stop processing immediately. The result contains the data lines processed up to, but not including, the data line where the exit condition occurred.

Syntax Rules

Use a colon (:) immediately following each conditional operator.

Use a semicolon (;) following the expression or equation to separate the statements.

Guidelines for Using Conditional Operators

- Use logical operators AND (&) and OR (,) to establish a condition with more than one part. For example, the following IF statement establishes a condition based on the values of two different fields:

  ```
  if:a < 100 & b < 100; ...
  ```

- When using multiple conditions, remember that once a condition is established with an IF statement, it remains in effect until the CAL function encounters another IF statement. You can include multiple IF/THEN/ELSE statements in your CAL equations; you are not limited to one.

- Use THEN and ELSE statements any number of times following a condition. The following example shows multiple THEN and ELSE statements:

  ```
  if:a = b;then:c = 1;then:d = 0;else:e = 1;else:f = 0
  ```

Processing Character Strings Using the Calculate Function

Use the following equation format and literal expression format to replace or compare character strings in a report:

receiving-label=literal-expression[; *(more equations)*]

where:

receiving-label When replacing strings this field is the field label or value label to hold the result of the expression. See also "Formulating Equations with the Calculate Function" in this section.

When comparing strings this field is the field label or value label to compare to. See also the example "Comparing Character Strings."

= When replacing strings this field is an equal sign to indicate a replace operation.

When comparing strings this field is any relational operator, not just an equal sign, to make the comparison.

literal-expression Literal data or character strings in one of the literal expression formats.

Literal Expression Formats

Table 4-6 lists and describes literal expression formats.

Table 4-6. Formats and Description of Literal Expressions

Format	Description
LIT(*x*)	Literal, or character equivalent, of label *x*.
LIT(*x*(*y*-*z*))	Literal, or character equivalent, of label *x*, starting at character position *y* for *z* characters.
"*data*"	"*Data*" can be up to 18 characters of literal data enclosed in either apostrophes or quotation marks. See the *Run Design Reference* for information on processing character strings in the CAL run statement.

A literal expression may have only a single item of literal data. Arithmetic and relational operators are not allowed.

Guidelines for Processing Character Strings

- In the LIT format, use any kind of label for label *x* except a multiple field label (unless it is the same as the receiving label).
- The function loads the receiving label starting at the leftmost character position and fills the remainder of the area with spaces if the specified literal is shorter than the area represented by the receiving label.

  If the literal is longer than the area represented by the receiving label, the function drops the rightmost characters of the literal.

  Note: *The size of the area represented by the receiving label is the size of the field if it is a field label, or 18 characters if it is a value label.*
- When performing literal comparisons, remember that before the CAL function compares the literal with the data in the label, it left-justifies them.

  If the label contains a different number of characters than the literal, the function adds spaces to the rightmost character positions to make the sizes equal.

  Unless you use the S(S) option, the function makes no distinction between uppercase and lowercase letters in the character strings compared.

Computing Dates and Times Using the Calculate Function

You can use the CAL function to perform computations on dates and times. The CAL function translates dates into a number of days relative to a fixed date, and times into a number of hours relative to midnight. All times are based on a 24-hour clock.

Date and Time Input and Output Formats

Use special input formats to indicate which date or time format the original data is in. The CAL function converts the date and time data to numeric values and processes them in your equations. After processing, the resulting date or time is formatted as you specify in special output formats.

Use the following date and time input formats to process dates and times in equations:

D*n*(*x*) and T*n*(*x*)

where:

D Date operation call.

T Time operation call.

n Number indicating the date or time format. See Table 4-6 for date and time formats.

(*x*) Numeric date or time value or any kind of label except a multiple field label (unless it is the same as the receiving label).

For example, this date input format specifies that dates in field A are in the format DD MMM YY (DATE2$ format): d2(a) .

Enter date and time output formats as equation options. In the equation, place a comma after the receiving label and follow it with the output format.

D(*n*) and T(*n*)

where:

D Date operation call.

T Time operation call.

(*n*) Number indicating the date or time format. See Table 4-6 for date and time formats."

For example, the following equation specifies that field A is to hold the result of a date computation in the format YYMMDD (DATE1$). The original date in field A was in the format DD MMM YY (DATE2$).

```
a,d(1)=d2(a)+90
```

Constant Labels

You can use the TODAY and TIME constant labels in your equations to process the current date and time:

- TODAY contains the current date expressed in number of days relative to a fixed date.
- TIME contains the current time expressed in the number of hours and fractions of hours relative to midnight.

See "Calculate Function Labels" in this section for more information on constant labels.

Using the W Option with Calculate

Use the W option to process dates by work week, where n is the number of days in the work week, 1 through 6. Default = 7 (assumed when you do not use the W option).

The function considers Monday as the beginning of the work week.

You can also use the W option as an equation option. See "Formulating Equations with the Calculate Function " in this section for more information on equation options.

For example with the W5 option, if six days were added to a date falling on a Monday, the resulting date would be Tuesday in the following week.

Guidelines for Computing Dates and Times

- Whenever you use date or time output formats, the system disables E, I, J, and R options since they also specify a type of output format.
- If the resulting date or time is not valid, the system fills the receiving field or label with asterisks (*) and assigns it a value of zero. If you intend to check for asterisks flagging invalid data, remember that value labels are 18 characters long and would contain 18 asterisks.
- The system right-justifies numeric formats and left-justifies alphabetic formats in the output field. It also fills any unused columns in the output field with spaces.

- To use minutes and seconds in time computations, convert them into hours (divide minutes by 60 and seconds by 3600).
- Date and time formats consist of a variable number of characters. The area represented by your receiving label must be large enough to hold the output format you specify. The size of each format is listed in the Minimum Size column in the chart on the following screen. In CAL run statements, note these sizes for receiving variables.
- If you specify a receiving label that is not large enough to hold the output format you specified, the system fills it with asterisks (*) and assigns it a value of zero.

Date and Time Formats

Table 4-7 lists formats for dates and times.

Table 4-7. Date and Time Formats

Format	Input	Output	Minimum Size
DATE0$ (YMMDD)	D0(x)	D(0)	5
DATE1$ (YYMMDD)	D1(x)	D(1)	6
DATE2$ (DD MMM YY)	D2(x)	D(2)	9
DATE3$ (YDDD)	D3(x)	D(3)	4
DATE4$ (YYDDD)	D4(x)	D(4)	5
DATE5$ (DDMMYY)	D5(x)	D(5)	6
DATE6$ (MM/DD/YY)	D6(x)	D(6)	8
DATE7$ (MONTH DD, YYYY)	D7(x)	D(7)	18*
DATE8$ (MMDDYY)	D8(x)	D(8)	6
TIME0$ (HH:MM:SS)	T0(x)	T(0)	8
TIME1$ (HH:MM)	T1(x)	T(1)	5
TIME2$ (HHMMSS)	T2(x)	T(2)	6
TIME3$ (HHMM)	T3(x)	T(3)	4
Month Name	n/a	D(C)	1**
Day	n/a	D(D)	2
Julian Day	n/a	D(J)	3
Month	n/a	D(M)	2
Day Number	n/a	D(N)	1
Day Name	n/a	D(W)	1**
Year	n/a	D(Y)	2
Hour Number	n/a	T(H)	2
Minute Number	n/a	T(M)	2
Second Number	n/a	T(S)	2

* DATE7$ fields must be exactly 18 characters wide; using fewer characters fills the field with asterisks.

** The number of characters for output formats C and W depends on the size of the receiving label.

Examples

Example 1: Computing Horizontally

This example multiplies the values in field B by .05, subtracts those values from the values in field A, and places the result values into field E. It also divides values in field D by 2 and places them back into field D.

```
vr.01
*Product.       .Retail. Sales .Space.Demo.            .
* Type  .       . $$$$ .Commiss. Req .Quan.Demo Results.
*=======. . . . .======.=======.=====.====.============.
********         ****** ******* ***** **** ************

                 a      b       c     d    e
e=a-(b*.05);d=d/2
```

v — Indicates that only equations that operated on valid data are to be included in the result.

r.01 — Indicates that the computation results are to be rounded to the nearest hundredth.

a, b, c, d, e — Field labels identifying the fields to process.

Example 2: Computing Horizontal Average, Sum, Minimum, and Maximum

This example works with this sample data:

```
* One . Two .Three. Four. Five. Six  .Seven . Eight .
*=====.=====.=====.=====.=====.======.======.=======.
     1     7     3  111
     2     6     6  222
     3     5     9  333
     4     4    12  444
     5     3    15  555
     6     2    18  666
     7     1    21  777
```

This is the function mask to use:

```
* One . Two .Three. Four. Five. Six  .Seven . Eight .
*=====.=====.=====.=====.=====.======.======.=======.
 ***** ***** ***** ***** ***** ****** ****** *******
 a     b     b     c     d     e      f      g
d=avg(b);e=sum(a,b,c);f=min(a,b,c);g=max(a,b,10)
```

The letters A, B, C, D, E, R, and G are field labels identifying the fields to process. Note that label B identifies two fields (multiple field label). The equation processes these fields and values in the following way:

d Contains an average of the B fields.

e Contains the sum of the A, B, and C fields.

f Contains the lowest (min) value of the A, B, or C fields.

g Contains the highest (max) value of either 10 or the value in the A or B fields.

This is the result of the calculations:

```
* One . Two .Three. Four. Five. Six  .Seven . Eight .
*=====.=====.=====.=====.=====.======.======.=======.
     1     7     3   111   5.0  122.0    1.0    10.0
     2     6     6   222   6.0  236.0    2.0    10.0
     3     5     9   333   7.0  350.0    3.0    10.0
     4     4    12   444   8.0  464.0    4.0    12.0
     5     3    15   555   9.0  578.0    3.0    15.0
     6     2    18   666  10.0  692.0    2.0    18.0
     7     1    21   777  11.0  806.0    1.0    21.0
```

Example 3: Totaling Fields Vertically

The following example shows the grand total of the Retail $$$$ field being created in two ways: using the vertical operator + and the built-in function VSUM.

Following is the function mask used in this example:

```
l
* Product . Sub .Produc. Whole . Retail .
*  Type   . Key . Cost . Sale$ .  $$$$  .
*=========.=====.======.=======.========.
********** ***** ****** ******* ********
                                + a
total=vsum(a)
```

l — Option indicating that all value labels are to be listed at the end of the result with their final values.

a — Field label identifying the field to process.

\+ — Vertical operator indicating the values in the Retail $$$$ field are to be totaled and the result placed beneath the field at the end of the result.

total= vsum(a) — Total is the value label that carries the vertical sum (VSUM) of field A.

Following is a portion of the result of the example:

```
* Product . Sub .Produc. Whole . Retail .
*  Type   . Key . Cost . Sale$ .  $$$$  .
*=========.=====.======.=======.========.
                 .
                 .
                 .
 GREENBOX7    I  14900   18625    26075
 GREENBOX8    J  15100   18875    26425
 GREENBOX9    K  15300   19125    26775
*                               --------
*                                 447300
.
. TOTAL   =            447300
                          ..... END REPORT .....
```

Example 4: Accumulating Sums Vertically

This example places the vertical cumulation of field A values into field B. This is a portion of the function mask used in this example:

```
j(c)
* Product .            .Space.  Demo    .                .
*  Type   .            . Req .Quantity. Demo Results    .
*=========.   . . .    .=====.========.===============.
**********              ***** ******** ***************
                         a             b
b=vsum(a)
```

j(c) Option indicating that commas are to be inserted every three digits in the result.

a, b Field labels identifying the fields to process.

Example 5: Comparing Fields

This example uses a conditional statement to test whether the values in field A equal those in field B. If they are equal, then the value 1 is placed in field C. If they are not equal, 0 is placed in field C. This is the data used in this example:

```
* One . Two .Three. Four. Five. Six  .Seven . Eight .
*=====.=====.=====.=====.=====.======.======.=======.
     1     7     3   111
     2     6     6   222
     3     5     9   333
     4     4    12   444
     5     3    15   555
     6     2    18   666
     7     1    21   777
```

This is the function mask used:

```
* One . Two .Three. Four. Five. Six  .Seven . Eight .
*=====.=====.=====.=====.=====.======.======.=======.
 ***** ***** ***** ***** ***** ****** ****** *******
 a     b                 c
if:a=b;then:c=1;else:c=0
```

This is the result obtained:

```
* One . Two .Three. Four. Five. Six  .Seven . Eight .
*=====.=====.=====.=====.=====.======.======.=======.
     1     7     3   111     0
     2     6     6   222     0
     3     5     9   333     0
     4     4    12   444     1
     5     3    15   555     0
     6     2    18   666     0
     7     1    21   777     0
```

Example 6: Comparing within a Range

This example does the following:

- The IF statement establishes a two-part condition: if the value in field A is greater than 150 and it is also less than or equal to 250, the equations defined in THEN statements are performed.
- Multiple THEN statements set field B to equal field A and field C to equal the value in field A times .5.
- Multiple ELSE statements are processed when the condition is not met. They set field B to 0 and field C to the value in field A times 2.

This is a portion of the function mask for the example:

```
* Product .           .Space.  Demo  .               .
*  Type   .           . Req .Quantity. Demo Results  .
*=========.    . . .  .=====.========.===============.
 *********             ***** ******** ***************

                       a     b        c
if:a>150 & a<=250;then:b=a;then:c=a*.5;else:b=0;else:c=a*2
```

Example 7: Conditionally Erasing Fields

This example does the following:

- The IF statement establishes the condition: if field A equals 0.
- The THEN statement uses an equation option, E, to indicate the field should be erased. The following THEN statement would also erase the fields:

  ```
  then:b=""
  ```

This is the function mask for this example:

```
*St.Status.By. Product .Serial.Produc.Order.Cust.Produc.
*Cd. Date .In.  Type   .Number. Cost .Numbr.Code. Plan .
*==.======.==.=========.======.======.=====.====.======.
 ** ****** ** ********* ****** ****** ***** **** ******
 b  b      b  b         a      b      b     b    b
if:a=0;then:b,e=0
```

Example 8: Conditionally Skipping Equations

In this example, the SKIP statement establishes the condition: if field A is equal to or greater than 14000. When this condition is true, the function skips the remaining equations and continues processing the next data line.

You can use the EXIT statement in place of the SKIP statement to stop processing all equations and display a result containing the lines processed up to, but not including, the line where the condition occurred.

This is the function mask for this example:

```
* Product . Sub .Produc. Whole . Retail . Sales .
*  Type   . Key . Cost . Sale$ .  $$$$  .Commiss.
*=========.=====.======.=======.========.=======.
 ********* ***** ****** ******* ******** *******
                 a      b       c        d
skip:a=>14000;d=c-b
```

Example 9: Replacing Character Strings

This example does the following:

- Replaces field A with the character string sh.
- Replaces the last character (4-1) of field B with the letter x. The notation (4-1) means to process one character beginning with the fourth column of the field.

Here is a portion of the function mask for the example:

```
*St.Order . Product .          .Sale.
*Cd.Number.  Type   .          .Rep .
*==.======.=========.   . . .  .====.
 ** ****** *********            ****
 a                              b
a="sh";b(4-1)="x"
```

Example 10: Moving Data between Fields

This example does the following:

- Replaces field D with literal data from columns 1 through 5 in field A. The notation (1-5) means to process five characters beginning with the first column of the field.
- Replaces field C with the literal data from field B.

Here is a portion of the function mask for the example:

```
* Product . Sub .           .  Demo  .               .
*  Type   . Key .           .Quantity. Demo Results  .
*=========.=====.   . . .   .========.===============.
 ********* *****             ** ***** ***************
 a         b                    c     d
d=lit(a(1-5));
c=lit(b)
```

Example 11: Comparing Character Strings

This example does the following:

- The first IF statement establishes the condition: if field C equals the character string dico.
- The first THEN statement replaces field B with the contents of field B plus 5.
- The next equation loads value label model with literal data from the first eight characters (1-8) in field A.

- The next IF statement establishes the condition: if the data in Model is not equal to the character string blackbox.
- The next THEN statement replaces field D with the character string jak.

Here is a portion of the function mask for the example:

```
*St.Order . Product .Ord.Cust.           .Sale.
*Cd.Number.  Type   .Qty.Code.           .Rep .
*==.======.=========.===.====.   . . .   .====.
 ** ****** ********* *** ****             ****
           a         b   c                d
if:c="dico";then:b=b+5;
model=lit(a(1-8));if:model<>"blackbox";then:d="jak";
```

Example 12: Searching for Character Strings and Numbers

The C option in this example indicates that only those data lines meeting the following two-part condition are to be included in the result: if field A equals 1 and if field B also equals the character string amco.

This is a portion of the function mask for the example:

```
c
*St.Ord.Cust.                 .               .
*Cd.Qty.Code.    Customer     .   Address     .
*==.===.====.=================.===============.
 ** *** **** ***************** ***************
    a   b
if:a=1&b="amco"
```

Example 13: Changing Line Types in a Report

This example does the following:

- The IF statement establishes the condition: if field A equals the character string sh.
- The multiple THEN statements do the following: Change the line type of those lines meeting the above condition to asterisk (*) — this is done with the LT constant label. Replace field B with the character string shx.

Here is a portion of the function mask for the example:

```
*St.Status.By. Product .         .Spc.
*Cd. Date .In.  Type   .         .Cod.
*==.======.==.=========.   . . . .===.
 ** ****** ** *********           ***
 a                                b
if:a="sh";
then:lt='*';
then:b='shx'
```

Example 14: Processing Based on Line Type

This example does the following:

- The A option indicates that all line types are to be processed.
- The first IF statement establishes the condition: if the line type currently being processed equals *.
- The first THEN statement replaces field A with the character string mauvebox.
- The next IF statement establishes the condition: if the line type currently being processed equals tab (|).
- The next THEN statement replaces field A with the character string whitebox.

Here is the function mask for this example:

```
a
*St.Status.By. Product .
*Cd. Date .In.  Type   .
*==.======.==.=========.
 ** ****** ** ********
               a
if:lt="*";then:a="mauvebox";
if:lt="|";then:a="whitebox";
```

Example 15: Subtotaling and Averaging on a Subkey Field

This example does the following:

- The FIRST statement initializes value label old to spaces.
- The IF statement establishes the condition: if the data in old equals the literal data in field A.
- The multiple THEN statements replace the contents of value label total with its current contents plus the value in field B and increases value label count by 1.
- The multiple ELSE statements set count to 1 and set total to the value in field B.
- The next equation replaces value label old with literal data from field A.
- The last two equations hold the running subtotals and averages: data in total is divided by count and placed in field C; field D is replaced with the data in total.

This is a portion of the function mask for this example:

```
*Product. Sub .            .Space.  Demo  .                .
* Type  . Key .            . Req .Quantity. Demo Results   .
*=======.=====.   . . .    .=====.========.===============.
 ******* *****              ***** ******** ***** *********
         a                   b              c     d
first:old="";
if:old=lit(a);then:total=total+b;then:count=count+1;
else:count=1;else:total=b;
old=lit(a);c=total/count;d=total
```

The final step in this example is to use the TOT function with O and S options against the CAL result to extract only those lines showing the final subtotals for each subkey group. Simply type the S parameter in the Sub Key field.

Example 16: Converting Date and Time Formats

This is the sample data used in this example:

```
*DATE7              DATE1  Results        TIME2  Results
*==================.======.==============.======.========.
 June 11, 1990      900611                130000
 June 26, 1990      900630                060000
```

This is the function mask used in this example:

```
*DATE7              DATE1  Results        TIME2  Results
*==================.======.==============.======.========.
 ****************** ****** ************** ****** ********
 a                         b              c      d
b,d(2)=d7(a);
d,t(0)=t2(c)
```

This is the result of this example:

```
*DATE7             DATE1  Results       TIME2  Results
*=================.======.=============.======.========.
 June 11, 1990     900611 11 JUN 90     130000 13:00:00
 June 26, 1990     900630 26 JUN 90     060000 06:00:00
```

Receiving labels B and D use equation options to specify the resulting format. D7 and t2 are input formats.

Example 17: Determining Day of the Week, Month Name, and Hour Number

This is the sample data used in this example:

```
*DATE7             DATE1  Results       TIME2  Results
*=================.======.=============.======.========.
 June 11, 1990     900611               130000
 June 26, 1990     900630               060000
```

This is the function mask used in this example:

```
*DATE7             DATE1  Results       TIME2  Results
*=================.======.=============.======.========.
 ****************** ****** ********* **** ****** ********
                   a      b         c   d      e
b,d(w)=d1(a);c,d(c)=d1(a);e,t(h)=t2(d)
```

This is the result of this example:

```
*DATE7              DATE1  Results        TIME2  Results
*==================.======.==============.======.========.
 June 11, 1990      900611 SATURDAY  JUN  130000       13
 June 26, 1990      900630 THURSDAY  JUN  060000        6
```

Receiving labels B, C, and E use equation options to specify the resulting format. D1 and t2 are input formats.

Example 18: Adding Days and Adding Hours

This is the sample data used in this example:

```
*DATE7              DATE1  Results        TIME2  Results
*==================.======.==============.======.========.
 June 11, 1990      900611                130000
 June 26, 1990      900630                060000
```

This is the function mask used in this example:

```
*DATE7              DATE1  Results        TIME2  Results
*==================.======.==============.======.========.
 ****************** ****** ************** ****** ********
 a                         b              c      d
b,d(2)=d7(a)+30;
d,t(2)=t2(c)+10
```

This is the result of this example:

```
*DATE7             DATE1  Results        TIME2  Results
*==================.======.==============.======.========.
 June 11, 1990     900611 11 JUL 90      130000 230000
 June 26, 1990     900630 26 JUL 90      060000 160000
```

Receiving labels B and D use equation options to specify the resulting format. D7 and t2 are input formats.

Example 19: Comparing Dates

This is the sample data used in this example:

```
*DATE7             DATE1  Results
*==================.======.==============.
 June 11, 1990     900611
 June 26, 1990     900630
```

This is the function mask used in this example:

```
*DATE7             DATE1  Results
*==================.======.==============.
 ****************** ****** **************
 a                 b      c
c=d7(a)-d1(b)
```

This is the result of this example:

```
*DATE7              DATE1  Results
*==================.======.==============.
 June 11, 1990      900611              0
 June 26, 1990      900630             -4
```

Field C contains the difference in days between field A and field B. D7 and d1 are input formats.

Note: *You can compare times in a similar way. Replace the date formats with time formats in the equation.*

Example 20: Comparing Dates Conditionally

This is the sample data used in this example:

```
*DATE7              DATE1  Results
*==================.======.==============.
 June 11, 1990      900611
 June 26, 1990      900630
 June 26, 1990      900622
```

This is the function mask used in this example:

```
*DATE7              DATE1  Results
*==================.======.==============.
 ****************** ****** ****   *******
 a                  b      c      d
c=d7(a)-d1(b);
if:c>0;then:d="Early";else:d="Late";
if:c=0;then:d="On time"
```

This is the result of this example:

```
*DATE7              DATE1  Results
*==================.======.==============.
 June 11, 1990      900611    0   On time
 June 26, 1990      900630   -4   Late
 June 26, 1990      900622    4   Early
```

- Field C contains the difference in days between field A and field B. D7 and d1 are input formats.
- The IF statements check if the difference in days in field C is greater than 0 or equal to 0.
- The THEN statements replace field D with various character strings depending on the contents of field C.

Example 21: Using the Current Date and Time in Comparisons

This is the sample data used in this example:

```
*DATE7              DATE1  Results        TIME2  Results
*==================.======.==============.======.========.
 June 11, 1990      900611                130000
 June 26, 1990      900630                060000
```

This is the function mask used in this example:

```
*DATE7              DATE1  Results        TIME2  Results
*==================.======.==============.======.========.
 ****************** ****** **** ********* ****** ********
 a                         b              c      d
b=d7(a)-today;
d,r.1=t2(c)-time
```

This is the result of this example:

```
*DATE7              DATE1  Results        TIME2  Results
*==================.======.==============.======.========.
 June 11, 1990      900611  152           130000     -1.1
 June 26, 1990      900630  167           060000     -8.1
```

- Field B contains the difference in number of days between field A and the current date.
- Field D contains the difference in number of hours between field C and the current time and rounds the result to the nearest 10th.
- D7 and t2 are input formats.

Example 22: Using a Constant Date and Time in Comparisons

This is the sample data used in this example:

```
*DATE7              DATE1  Results        TIME2  Results
*==================.======.==============.======.========.
 June 11, 1990      900611                130000
 June 26, 1990      900630                060000
```

This is the function mask used in this example:

```
*DATE7              DATE1  Results        TIME2  Results
*==================.======.==============.======.========.
 ****************** ****** ***            ****** ********

                    a      b              c      d
b=d1(900603)-d1(a);
d=t2(c)-t3(0500)
```

This is the result of this example:

```
*DATE7              DATE1  Results        TIME2  Results
*==================.======.==============.======.========.
 June 11, 1990      900611  -8            130000        8
 June 26, 1990      900630 -27            060000        1
```

Fields B and D contain the differences specified. D1, t2, and t3 are input formats.

Example 23: Adding Days to a Date Using a Five-Day Workweek

This is the sample data used in this example:

```
*DATE7              DATE1  Results
*==================.======.==============.
 June 11, 1990      900611
 June 26, 1990      900630
```

This is the function mask used in this example:

```
w5
*DATE7              DATE1  Results
*==================.======.==============.
 ****************** ****** **************
 a                  b      c
c,d(2)=d7(a)+10
```

This is the result of this example:

```
*DATE7              DATE1  Results
*==================.======.==============.
 June 11, 1990      900611 24 JUN 90
 June 26, 1990      900630 08 JUL 90
```

Receiving label C uses an equation option to specify the resulting format. D2 and d7 are input formats.

Example 24: Computing the Difference in Workdays, Method 1

Note: *Since a weekend date has the same relative workday number as the previous workday, some inconsistencies can occur when you calculate the difference in workdays.*

This example calculates the difference in workdays not including the first day (if it is a workday) but including the last day (if it is a workday).

This is the sample data used in this example:

```
*DATE7              Results
*==================.==============.
 June 11, 1990
 June 26, 1990
 June 28, 1990
```

This is the function mask used in this example:

```
*DATE7              Results
*==================.==============.
 ****************** **************
 a                  b
b,w5=d7(a)-d1(900603)
```

This is the result of this example:

```
*DATE7             Results
*=================.=============.
 June 11, 1990                 5
 June 26, 1990                15
 June 28, 1990                17
```

- Receiving label B uses the W5 equation option to indicate five days in the workweek.
- The date 900603 is subtracted from field A. (Dates in field A must be later than 900603 for this equation to work as expected.) D7 and dl are input formats.

Note: *This method counts the difference between Monday and Friday as four days (day 5 minus day 1).*

Example 25: Computing the Difference in Workdays, Method 2

Note: *Since a weekend date has the same relative workday number as the previous workday, some inconsistencies can occur when you calculate the difference in workdays.*

This example calculates the difference in workdays including both the first and last day (if they are workdays).

This is the sample data used in this example:

```
*DATE7             Results
*=================.=============.
 June 11, 1990
 June 26, 1990
 June 28, 1990
```

This is the function mask used in this example:

```
*DATE7              Results
*==================.==============.
 ****************** **************
 a                  b
wkday,d(n)=d1(900603);b,w5=d7(a)-d1(900603)+(wkday<=5)
```

This is the result of this example:

```
*DATE7              Results
*==================.==============.
 June 11, 1990                   6
 June 26, 1990                  16
 June 28, 1990                  18
```

- Loads value label wkday with the day number of the date 900603 (Monday = 1, Tuesday = 2, and so on).
- Subtracts dates as specified. (See method 1.)
- The relational expression wkday<=5 produces a value of 1 if it is true (wkday is a workday) or 0 if it is false. This function adds the value to the difference.

Note: *This method counts the difference between Monday and Friday as five days (day 5 minus day 1 plus 1 if the earlier date is a workday).*

Example 26: Computing the Difference in Workdays, Method 3

Note: Since a weekend date has the same relative workday number as the previous workday, some inconsistencies can occur when you calculate the difference in workdays.

This example calculates the difference in workdays the same way as the Date function, as follows:

- If the first day is not a workday, the function moves it backward to the closest workday.
- If the last day is not a workday, the function moves it forward to the closest workday.
- If both the first and last days are not workdays, the function moves both of them forward to the next workday.
- When the difference is calculated, the function does not include the first day; Friday to Saturday is one workday and Sunday to Monday is one workday.

This is the sample data used in this example:

```
*DATE7              Results
*==================.==============.
 June 11, 1990
 June 26, 1990
 June 28, 1990
```

This is the function mask used in this example:

```
*DATE7              Results
*==================.==============.
 ****************** **************
 a                  b
wkday,d(n)=d1(900603);wkday1,d(n)=d7(a);
b,w5=d7(a)-d1(900603)+((wkday<=5)&(wkday1>5))
```

This is the result of this example:

```
*DATE7              Results
*=================.==============.
 June 11, 1990                 6
 June 26, 1990                16
 June 28, 1990                17
```

- Loads value label wkday with the day number of the date 900603 (Monday = 1, Tuesday = 2, and so on).
- Loads value label wkday1 with the day number of the date in field A.
- Subtracts dates as specified. (See method 1.)
- The relational expression wkday<=5 produces a value of 1 if it is true (Wkday is a workday) or 0 if it is false.
- The relational expression wkday1>5 produces a value of 1 if it is true (Wkday1 is a weekend day) or 0 if it is false.
- The logical AND (&) specifies that only when both expressions are true will a value of 1 be added to the difference. If one or both of the expressions are false, the resulting value is 0. See the "Relational Expressions" in this section for information on AND and OR truth conditions.

CHG (Change)

Use the Change (CHG) function to find and replace character strings anywhere within one report and to create a result containing both changed and unchanged lines. (You can use the O option to create a result containing only changed lines.)

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Was a report on display?

   Yes: Fill in the function mask.

   No: Fill in the function form then the function mask.

Menu Path

Select Task
    Modify Report
        Locate and Change

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

CHG [*rdc f*];/*tgtstr*/*replstr*/[*o*]

Changes the target string to the replacement string, regardless of line type or column position in the current format.

CHG

Was a report on display?

Yes: Displays the function mask.

No: Displays the function form.

CHG *rd*[*c f*]

Displays the function mask.

continued

Control Line Procedure (cont.)

Format Fields	*rdc*	Report to process.
	f	Report format in which to search for data (lets you look for data in fields beyond column 80 if you do not already have those columns on display).
	;	Required character.
	/	Delimiter identifying the character string boundaries. Can be any character except those used in the target or replacement string, or a caret (^).
	tgtstr	Character string to be located.
	replstr	Replacement string. This string does not have to contain the same number of characters as the target string.
	o	Options. In this function format, the A, F, and M options are always assumed.

Options

A Processes all line types, ignoring the character in column 1 of the function mask. Changes the line type of all lines containing finds to the first character of the replacement string.

When you specify the target and replacement strings using the control line procedure, the A, F, and M options are always assumed.

C Distinguishes between uppercase and lowercase letters.

1100: On OS 1100 MAPPER Systems, this option applies only to full character set (FCS) reports.

continued

Options (cont.)

F

Processes all line types and uses the first column under the function mask as part of the target and replacement strings.

When you specify the target and replacement strings using the control line procedure, the A, F, and M options are always assumed.

1100: On OS 1100 MAPPER Systems, this option processes all line types and locates and changes the entire string. It does not locate strings that start in column one.

When you specify the target and replacement strings using the control line procedure or when you use the LCH run statement, the A, F, and M options are always assumed.

M

Designates the first character of the target string as the line type designator. Used primarily in runs to locate specific line types. When you specify the target and replacement strings using the control line procedure, the A, F, and M options are always assumed.

O

Creates a result containing each line on which strings were changed.

OU

Creates an update result containing each line on which strings were changed. This option can be used only with reports, not with results.

S

Starts the search at the first line on the screen.

S*x*

Starts the search at line *x*, where *x* is a positive number.

S*x*-*y*

Searches lines *x* through *y*.

S*x*,*n*

Starts the search at line *x* and searches *n* lines.

T*x*

Specifies any character as a wildcard character. Default wildcard character = blank. See the examples in this section. Do not use the wildcard character as the first character in the character string.

1100: On OS 1100 MAPPER Systems, if the replacement string is shorter than the target string, the rightmost characters are filled with the specified transparent character.

Outcome

- A result is displayed containing all the lines of the report. In the result, the new string replaces the target string at each occurrence.

  If you used the O option, the result contains only the lines where the target string was found and changed. The OU option creates an update result.

- If the specified data is not found, an unchanged result is displayed.

- If the replacement string is larger than the target string, the CHG function shifts the remaining characters visible in the current format to the right on the lines where changes were made. If the replacement string is smaller than the target string, the characters move to the left on the line.

Guidelines

Completing the Function Mask

To fill in the function mask, follow these steps:

1. On the first line under the mask, type the line type to process in the first column, followed by the target.
2. Press the **Erase to End of Line** key.
3. Move the cursor to the next line.
4. Type the line type to process in the first column followed by the replacement string.
5. Press the **Erase to End of Line** key.
6. Move the cursor to the next line and transmit.

Changing a String in a Specific Field

To change a string in a specific field, erase all asterisks in the function mask except those in the field you want to change.

Changing a String in Freeform Lines

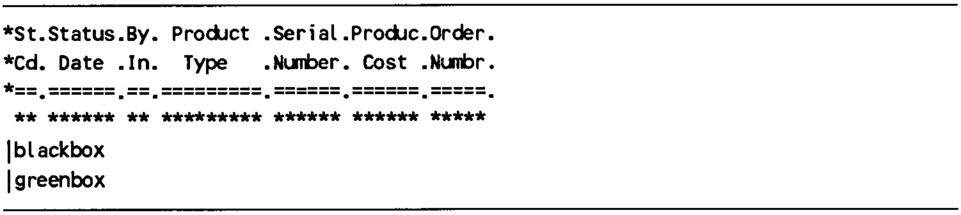
The fastest way to find a string in freeform lines is to use the following format with a report on display:

```
CHG ;/tgtstr/replstr/
```

Examples

Example 1

This example replaces blackbox with greenbox in any of the fields. You would erase the rest of the tabs on the blackbox line with the **Erase to End of Line** key.



```
*St.Status.By. Product .Serial.Produc.Order.
*Cd. Date .In.  Type   .Number. Cost .Numbr.
*==.======.==.=========.======.======.=====.
 ** ****** ** ********* ****** ****** *****
|blackbox
|greenbox
```

Example 2

Changing the character string ip to sh in report 2B0:

```
chg 2b0;/ip/sh/
```

Example 3

Changing the character string black to white in the current report, using the S option to process only line 20 to the end of the report:

```
chg ;/black/white/s20
```

Example 4

This example changes the string r m (the letter R, two spaces, and the letter M) to the string room in report 2B0. Because the spaces in the character string are wildcards, possible target strings might be ream, roam, and so on.

```
chg 2b0;/r  m/room/
```

Example 5

This example changes all dates in DATE1$ format (YYMMDD) where the year 1990 is filled in along with any month number, but no day (where there are spaces in the DD portion of the date) to the first of the month. The T option changes the wildcard character to a dollar sign ($) so that the request can find spaces.

```
chg ;/90$$  /90$$01/t$
```

CMP (Compare Data)

Use the Compare Data (CMP) function to compare data in two reports. It performs a character-to-character comparison of specified fields in the two reports.

Prerequisite

One of the reports must be on display.

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Was a report on display?

   Yes: Fill in the function mask.

   No: Fill in the function form then the function mask. For help while filling in the function form, press the **Help** key.

Menu Path

Select Task
  Compare Reports
    Compare Two Reports

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

CMP
Displays the function form.

CMP rd[*c f,l*]
Displays the function mask.

Format Fields

rdc Report to compare.

f Report format in which to compare data (lets you compare data in fields beyond column 80 in the second report).

l Line number of the reports where the comparison is to start.

Options

A	Processes all line types.
C(S)	Distinguishes between uppercase and lowercase letters. **1100:** On OS 1100 MAPPER Systems, this option applies only to full character set (FCS) reports.

Parameters

1 - 25	Specifies the sequence of fields to compare, beginning with 1. In the function mask, type each parameter in the first column of the field. Erase or add asterisks in the function mask, if necessary, to make the fields you are comparing the same size. ***Note:*** *Maximum field size is 255 characters. See also Guidelines in this section.*

Outcome

If the function detects a difference, it redisplays the report you had on display with the cursor located in the column where the difference occurs. A message states that a difference was found.

Otherwise, a message states that the fields are identical.

Guidelines

Continuing the Compare Operation

To continue comparing fields in the report, restart the CMP function. In the CMP request, include the line number at which the comparison should start.

Comparing Whole Lines

Since you can compare fields that are up to 255 characters wide, you can compare whole lines of data. (Use the CMP run statement in a run to compare lines longer than 80 characters.)

Add asterisks in the function mask to make a solid line of asterisks. Use the A option if you want to process all line types.

Example: Comparing All Line Types

This example compares two reports. If the function detects a difference, it displays the report you had on display at the line where it found the difference. If it detects no difference, it displays a message stating that the fields are identical.

Here is the function mask for this example:

```
* Produc . Sub .Produc. Whole .
*  Type  . Key . Cost . Sale$ .
*========.=====.======.=======.
 ******** ***** ****** *******
 1              2

*St.Status.By. Produc .Serial.Produc.
*Cd. Date .In.  Type  .Number. Cost .
*==.======.==.========.======.======.
 ** ****** ** ******** ****** ******
              1               2
```

1 and 2 Identify the fields to compare.

CNT (Count)

This section applies only to the OS 1100 MAPPER System.

Use the Count (CNT) function to analyze and summarize data. Based on key fields, the function computes subtotals, percentages, averages, entry counts, and more.

The CNT function has several advantages over other MAPPER functions. These include the ability to do the following:

- Subtotal fields without having to sort them first.
- Perform calculations on a single report or across an entire drawer.
- Process more than one distinctly different set of analyses with one function request.
- Reduce processing time in two ways. You can replace a sequence of two or more other functions with the CNT function. The function scans the report in one pass no matter how many analyses you request.

The following tasks are just a few of the many operations you can perform with the CNT function:

- Count the number of unique entries in key fields.
- Calculate the subtotal, average, minimum, or maximum value for each unique key.
- Compute the percentage of entries in a certain key of the total number of entries.
- Compute the percentage that each subtotal field is of the grand total.
- Numerically round each requested field separately.
- Perform calculations based on numeric, date, or time intervals.
- Display only specified fields in the result.

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Was a report on display?

   Yes: Fill in the function mask.

   No: Fill in the function form then the function mask. For help while filling in the function form, press the **Help** key.

Menu Path

Select Task
    Perform Math Operations
        Count and Analyze Data

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

CNT

Was a report on display?

Yes: Displays the function mask.

No: Displays the function form.

CNT *rd*[*c f*]

Displays the function mask.

Format Fields

rdc Report to process.

f Report format in which to process data.

Options

A Processes all line types.

B Extends the boundaries for scaling intervals to the minimum and maximum values defined in the scaling option. Every interval between those boundaries is displayed, even intervals with no entries.

C Distinguishes between uppercase and lowercase letters. Default = the system processes and displays the keys as uppercase.

D[*n*] Processes only those keys that occur *n* or more times, where *n* is an integer greater than 1. Default = 2.

E Extracts result fields from the last occurrence of each unique key. Default = first occurrence.

F Extracts result fields from the first occurrence of each unique key. When using a minimum (<) or maximum (>) operator. Default = minimum or maximum field.

H Displays only the first set of headings in the result when you process more than one set of parameters in a single function request.

I Includes out-of-range keys. Substitutes the minimum or maximum boundary value, when applicable, for an out-of-range key. The key field is then considered valid, and all its values are counted. Default = the line is discarded for values outside the boundaries.

N*n* Substitutes invalid numeric data with the value *n*. Default = 0.

O Omits data lines from the result. Only heading lines, grand total summary, and warning messages are included in the output result. The T and W options are automatically implied when the O option is specified.

P Removes all nonreferenced fields from the result and reorders the fields, from left to right, according to this hierarchy:

1. Key fields in numeric order
2. Extraction fields in alphabetical order
3. Computational fields (subtotals, averages, and so on)
4. Minimum fields
5. Maximum fields
6. Entry counts
7. Percentages

continued

Options (cont.)

R*x*-*y* Processes a range of reports from *x* through *y*.

R*x*,*y* Processes reports *x* and *y*. If you use this option in the CNT run statement, place the comma within apostrophes (','). The reports are processed in the order you specify.

R*x*-*y*,*z* Processes reports *x* through *y* and also *z*. The reports are processed in the order you specify.

S*n* Creates intervals or scales in key fields.

S*n*(*intv*[/*min*/*max*])	Scales numeric values.
S*n*D*f*(*intv*[/*min*/*max*/*strt*])	Scales dates.
S*n*T*f*(*intv*[/*min*/*max*/*strt*])	Scales times.

n Number of the key field to scale (1 through 9).

f Date format number.

0	YMMDD	3	YDDD	6	MM/DD/YY
1	YYMMDD	4	YYDDD	7	MONTH DD,YYYY
2	DD MMM YY	5	DDMMYY	8	MMDDYY

Time format number.

0	HH:MM:SS	2	HHMMSS
1	HH:MM	3	HHMM

intv Interval size.

Date intervals:

*n*D *n* = number of days (default)
*n*W *n* = number of weeks
*n*M *n* = number of months
*n*Y *n* = number of years

continued

Options (cont.)

Time intervals:

*n*S *n* = number of seconds
*n*M *n* = number of months
*n*H *n* = number of hours

min Minimum value at which to begin scaling.

max Maximum value at which to end scaling.

strt Starting value.

If *intv* is:	Then *strt* value is:	Example:
*n*D or *n*W	Three-character day name	SUN
*n*M	Three-character month name	MAY
*n*Y	Two- or four-digit year	90 or 1990
*n*S,*n*M,*n*H	Same format as key field	

T Displays a total for each computed field. For example, subtotaling shows a grand total; averaging shows a grand average; entry counting shows total entries; minimum and maximum operators show the smallest or largest values in the field. You can also calculate the overall standard deviation of values in a field.

U Displays the upper limit of the interval for scaled results rather than the lower limit.

V Does not count lines with invalid numeric, date, or time fields. Default = 0 for numeric operations; January 1, 1944, for date calculations; or 00:00:00 for time operations.

W Adds messages at the end of the result showing the number of lines skipped due to invalid key or data fields.

Z Displays all intervals in the result, even those with no entries. This option allows you to create a scale with no gaps. Use this option only with the S option.

* Flags invalid subtotals, averages, standard deviations, maximums, and minimums if invalid data were used in the calculations.

Parameters

1-9	to specify key fields (at least 1 is required)
N	to specify numeric key fields (1N)
D*f*	to specify date key fields (See the S*n* option for formats.) *f* = 0-8 to specify date format Example: 1D0, specifies key field 1 in date format 0
T*f*	to specify time key fields (See the S*n* option for formats.) *f* = 0-3 to specify time format Example: 2T3, specifies key field 2 in time format 3
R	to set up report key fields Example: 1R, loads the field with the report number when processing a drawer or range of reports
L	to specify line key fields Example: 1L, loads the field with the input report line number

Operators

=	Entry count
%	Percentage
+	Subtotal
+%	Subtotal percentage
/	Average
<	Minimum
>	Maximum
!-	Sample standard deviation
!+	Population standard deviation

continued

Operators (cont.)

!!-	Sample variance
!!+	Population variance
:R	Places the input report number into the field.
:L	Loads the field with the input report line number.
:M	Loads the field with the corresponding analysis.

Outcome

- A result is displayed with the completed calculations.
- The result contains a one-line summary of each unique key.
- The result is sorted by key field(s) in ascending order.
- All nonreferenced fields are filled with spaces.
- When an invalid key is encountered in the input report, the entire line is omitted from calculations. When invalid data is encountered in a nonkey field, the value is considered to be zero for numeric operations; January 1, 1944, for date calculations; or 00:00:00 for time operations.

Guidelines

How the CNT Function Works

The CNT function groups identical values in the key field together and performs calculations for each group, or key.

Specifying Parameters

Specify key fields by placing parameters, 1 through 9, in the appropriate fields in the function mask. You must specify at least one key field for each set of parameters.

Specifying Fields to Operate On

Specify fields to operate on by placing parameters in appropriate fields in the function mask. For example, put an equal sign (=) in the field you want to load with the number of times a unique key value occurs.

Key Field Parameters

Key fields contain data to be grouped together. You perform calculations on fields based on the key parameters you enter in the key field.

Key field parameters are the numbers 1 through 9, where 1 is the most significant. You can begin with a number other than 1. Always specify at least one key field for each set of parameters. The following are types of key fields.

Alphanumeric key fields are for specifying character data. Use a single digit, 1 through 9. By default, upper and lowercase characters are considered equal.

Numeric key fields are for specifying numeric fields. Use a key number and the letter N, for example, 1N. Numbers can be left or right justified.

Date and time key fields are for specifying date or time fields. Use a key number, the letter D or T, and a date or time format number. For example, 1T0 indicates a key field containing times in format 0, HH:MM:SS. See the S*n* option for the date and time format numbers.

Constant-Label Key Field Parameters

Unlike key fields that refer to the contents of a report field, constant-label key fields ignore field data. Instead, they represent execution-time values.

Use report key parameters only when processing an entire drawer or range of reports. With report keys, you use the report number as a key and perform computations based on each report. Specify a report key with a parameter containing a key number and the letter R; for example, 1R. The key field is loaded with the report number, and changes with each processed report.

Use line keys to match output and input lines. Specify a line key with a key number and the letter L, for example, 1L. The line key field is loaded with the input report line number, and changes with each processed line.

Constant Label Operators

Constant labels (:R, :L, and :M) load fields with constants whose values are based on conditions that exist while the function is executing.

The report number label operator (R:) places the input report number into the field.

The line number label operator (:L) loads the field with the input report line number.

The multiple analysis number label operator (:M) loads the field with the corresponding analysis number. If you use more than one set of parameters (perform more than one analysis) in a function request, the function creates a single result containing output from all of the analyses. This operator (:M) loads the specified field with the number of the analysis that is the source of the output line. For example, output lines created by the first set of parameters are loaded with a value of one.

Extraction Parameters

Ordinarily, only key fields and fields involved in operations are included in a CNT result. If you use extraction parameters, however, you can include the contents of other fields as well. An extraction parameter is a single letter placed in a field. Do not label more than one field with the same letter. By default extraction fields are loaded from the report line where each unique key occurs the first time.

The exclusion parameter is a minus sign following the extraction parameter and indicates that the field should be excluded from the result. Use this operator only with labeled fields. For example, A- labels a field as A, but does not include the contents of the field in the result.

Scaling: Arranging Keys into Fixed-Size Intervals

Scaling creates intervals of values within the key field. For example, you can scale the values in a date key field to perform calculations based on weekly intervals. To scale a key field, specify the appropriate scaling (S) option.

Scaling Guidelines

- Each scaled key field must be defined by its own scaling option.
- Each interval shown in the result has one output line associated with it.
  - By default, the lower value in the interval is displayed.
  - By default, only intervals containing values are included.

Scaling Types

You can scale three types of data: numbers, dates, and times. Each scaling type uses a different format for the S option.

Numeric Scaling

Numeric scaling groups values together within the same numeric interval and treats them as a single key. Following is the format to use for numeric scaling:

S*n*(*intv*[/*min*/*max*])

where:

S	Option letter.
n	Number of the key field to be scaled.
intv	Size of the interval.
min	Minimum scaled value. Key field values smaller than this value are ignored.
max	Maximum scaled value. Key field values larger than this value are ignored.

Date Scaling

By scaling date keys, you group dates together within the same interval and treat them as a single key. Following is the format to use for date scaling:

SnDf(intv[/min/max/strt])

where:

S Option letter.

n Number of the key field to be scaled.

D Indicator of date scaling

f Date format number. See the S*n* option for the format numbers.

intv Size of the interval. This field consists of a positive integer number (n) and a letter describing the type of interval. Formats of the field follow:

*n*D *n* = number of days (default)
*n*W *n* = number of weeks
*n*M *n* = number of months
*n*Y *n* = number of years

Example: 2M creates an interval size of two months.

min Minimum scaled date in the same date format as the data in the key field. Earlier dates are ignored.

max Maximum scaled date in the same date format as the data in the key field. Later dates are ignored.

strt Interval starting date. Must be same date format as the key field, or must be a special value listed here. The *strt* value depends on the *intv* value.

If intv is:	Then strt value is:	Example
*n*D or *n*W	Three-character day name	SUN
*n*M	Three-character month name	MAY
*n*Y	Two- or four-digit year	90 or 1990

For example, if you create weekly intervals and choose a starting point of WED, weeks begin on Wednesday and end on Tuesday.

Defaults = January 1, 1944, (for year or month scaling); January 2, 1944, Sunday (for day or week scaling).

Time Scaling

By scaling time keys, you group times together within the same interval and treat them as a single key. Following is the format to use for numeric scaling:

S*n*T*f*(*intv*[/*min*/*max*/*strt*])

where:

S Option letter.

n Number of the key field to be scaled.

T Indicator of time scaling

f Time format number. See the S*n* option for the format numbers.

Intv This field consists of a positive integer number (n) and a value describing the type of interval. Formats of the field follow:

*n*S = *n* number of seconds (default)
*n*M = *n* number of minutes
*n*H = *n* number of hours

For example, 2M creates an interval size of two minutes.

min Minimum scaled time in the same time format as the data in the key field. Times in the key field that are earlier than this time are ignored.

max Maximum scaled time in the same time format as the data in the key field. Later times are ignored.

strt Starting time for intervals. This field must be in the same time format as the data in the key field. Default = 00:00:00.

Examples

This sample report is used in CNT function examples in the following examples. It lists the location, date, and time of systems that have gone down. It also lists the number of hours that the systems were down, the cause code, and the cost of the down time.

```
*Location. Date .  Time  .Down. Cause.  Value      Cost   .Count.  %
*========.======.========.====.======.==========.=========.=====.=====
 Chicago  890221 05:00:32  2.2  CODE1                  550
 Boston   890424 03:00:45   .5  Code3                   40
 Chicago  890607 09:00:23  1.3  Code2                 2900
 New York 890607 11:47:47  1.0  Code2                   10
 Boston   890708 14:31:02  0.5  Code1                   55
 Boston   890708 08:24:01  0.1  Code2                    9
 Chicago  890930 18:45:02  4.0  Code3                 7500
 Chicago  891130 23:24:00  0.7  Code2                   20
 Boston   891202 03:05:09    1  Code2                  900
```

Example 1: Counting Keys and Calculating a Percentage

This example counts the times each unique key appears in the key field and calculates the percentage of total down-time events for each location.

This is a portion of the function mask used in this example:

```
*Location.Date  .Time  .Down. . . .Count.  %
*========.======.======.====. . . .=====.=====
 1                                  =     %
```

1 Indicates the key field.

= Identifies where to place the calculation of the number of times a city had down time.

% Identifies the where to place the percentage of times the city had a down-time event.

This is the result of the calculation:

```
*Location.Date  .Time  .Down. . . .Count.  %  .
*========.======.======.====. . . .=====.=====.
 BOSTON                                 4  44.4
 CHICAGO                                4  44.4
 NEW YORK                               1  11.1
```

Example 2: Subtotaling and Averaging

This example subtotals the number of hours each system was down for each cause code and averages the cost for each cause. This is a portion of the function mask used in this example:

```
*Location. . . .Down. Cause.  Value    .  Cost   . . .
*========. . . .====.======.==========.=========. . .
                +     1                /
```

1 Indicates the key field.

+ Subtotals the amount of down time for each cause.

/ Calculates the average cost of each cause's down-time.

This is the result of the calculation:

```
*Location. . . .Down. Cause.  Value      Cost   . . .
*========. . . .====.======.==========.=========. . .
                2.70  CODE1            302.50000
                4.10  CODE2            767.80000
                4.50  CODE3                 3770
```

Example 3: Rounding

This example shows how to round each result field. Enter an R in the first column of the line below the parameters. Enter the rounding value under each field to round. Use the same rounding values as for the Calculate (CAL) function. This example builds on the previous example of subtotaling and averaging by rounding subtotal values to the nearest tenth and by rounding average values to the nearest hundredth.

This is a portion of the function mask used in this example:

```
*Location. . . .Down. Cause.  Value    .  Cost   . . .
*========. . . .====.======.==========.=========. . .
                +    1                  /
R               .1                     .01
```

This is the result of the calculation:

```
*Location. . . .Down. Cause.  Value    .  Cost   . . .
*========. . . .====.======.==========.=========. . .
                 2.7  CODE1                302.50
                 4.1  CODE2                767.80
                 4.5  CODE3               3770.00
```

Example 4: Vertical Cumulation

To produce a vertical cumulation, add a plus sign (+) following another operator in that field.

This is a portion of the function mask used in this example:

```
*Location. . . .Down. Cause.  Value   .  Cost   . . . .
*========. . . .====.======.==========.=========. . . .
  1            =    =+     %          %+
```

1 Indicates the key field.

= Specifies the field in which to display the number of times each unique key occurs.

=+ Specifies the field in which to display the vertical cumulative entry count of the keys.

% Calculates the percent of the total cost for each unique key.

%+ Specifies the field in which to place the vertical cumulative percent cost of each unique key.

Following is the result of the previous vertical cumulations.

```
*Location. . . .Down. Cause.  Value   .  Cost   . . .
*========. . . .====.======.==========.=========. . .
 BOSTON           4      4   44.44444  44.44444
 CHICAGO          4      8   44.44444  88.88889
 NEW YORK         1      9   11.11111 100.00000
```

The result indicates the following:

- Chicago experienced 4 down-time events.
- The average cost of a down-time event in New York was 11.11111.
- Boston and Chicago together made up 88.888889 percent of the total cost for down times.
- For all three locations, systems went down a total of 9 times.

Example 5: Placing Result Values in a Selected Field

This example uses extraction parameters to place result values in fields you select. This is a portion of the function mask used in this example:

```
*Location. . . .Down. Cause.  Value    .  Cost   . . . .
*========. . . .====.======.==========.=========. . . .
 1              A           +A
```

1 Indicates the key field.

A Labels the field as A and extracts data for display. The value displayed in the Down field is the first value associated with the key in the input report.

+A Calculates Down field subtotals (labeled as A).

This is the result of the calculation:

```
*Location. . . .Down. Cause.  Value    .  Cost   . . . .
*========. . . .====.======.==========.=========. . . .
 BOSTON            .5       2.10000000
 CHICAGO          2.2       8.20000000
 NEW YORK         1.0                1
```

Note: If A- (a label and the exclusion parameter) had been placed in the Down field, the values in that field would have been excluded from the result.

Example 6: Performing Multiple Operations on a Single Field

The following example demonstrates how to perform multiple operations on one field. Following is a portion of the function mask:

```
*Location. . . .Down. Cause.  Value   .  Cost   . . . .
*========. . . .====.======.==========.=========. . . .
 1              A+         /A
```

1 Indicates the key field.

A+ Labels the Down field as A, calculates the total down time for each Location, and places the subtotals in the Down field.

/A Calculates the average down time for each location and places the values in the Value field.

This is the result of the calculation:

```
*Location. . . .Down. Cause.  Value   .  Cost   . . . .
*========. . . .====.======.==========.=========. . . .
 BOSTON         2.10       0.52500000
 CHICAGO        8.20       2.05000000
 NEW YORK          1                1
```

Example 7: Finding Percentages of Grand Totals

This example shows how to find the values for certain fields as percentages of the grand total. Following is a portion of the function mask:

```
*Location. . . .Down. Cause.  Value   .  Cost   . . . .
*========. . . .====.======.==========.=========. . . .
                1              +%                      %+
r              .1                       1
```

1 Indicates the key field.

+% Computes, for each location, the number of hours down expressed as a percent of the grand total hours down.

%+ Calculates, for each location, the number of down-time events expressed as a percent of total number of down times. Displays the cumulative percent.

r Rounds result fields as specified (Down is rounded to tenths [.1], Cost is rounded to units [1]).

This is the result of the calculation:

```
*Location. . . .Down. Cause.  Value   .  Cost   . . . .
*========. . . .====.======.==========.=========. . . .
 BOSTON         18.6                          44
 CHICAGO        72.6                          89
 NEW YORK        8.8                         100
```

Example 8: Performing Multiple Analyses in a Single Pass

Using more than one set of parameters (performing more than one analysis) in a single function request creates a single result containing output from each analysis. You can specify up to eight sets of parameters and analyses.

Following is a portion of a function mask that demonstrates how to produce this result:

```
*Location. . . .Down. Cause.  Value   .  Cost   . . . .
*========. . . .====.======.==========.=========. . . .
 1              A/                     +A
 =                   1
```

First set of parameters:

1 Indicates the key field.

A/ Labels the Down field as A, calculates the average hours down for each location, and places the values in the Down field.

+A Calculates the total number of hours down for each location and places the values in the Cost field.

Second set of parameters:

= Counts the number of times each system was down. In other words, counts the number of times each unique key appears in the key field.

1 Specifies the key field.

This is the result of the previous calculation. Note that the .EJECT command appears between each analysis output.

```
*Location. Date .  Time  .Down. Cause.  Value      Cost   .Count.  %
*========.======.========.====.======.==========.=========.=====.=====
 BOSTON                   0.53                   2.1000000
 CHICAGO                  2.05                   8.2000000
 NEW YORK                    1                           1
.EJECT
.DATE 03 MAY 90  20:10:02  RID     21A   03 MAY 90  NEWUSER
.Count Practice Report                                    BY: MAPCOORD
*Location. Date .  Time  .Down. Cause.  Value      Cost   .Count.  %
*========.======.========.====.======.==========.=========.=====.=====
        2                       CODE1
        5                       CODE2
        2                       CODE3
                     ..... END REPORT .....
```

Example 9: Scaling a Numeric Key Field

Following is an example that demonstrates how to specify intervals and group data within them. This is a portion of the function mask:

```
s1(.5)
*Location. . . .Down. Cause.  Value    .  Cost   .
*========. . . .====.======.==========.=========.
                 1                       /
r                                        1
```

This is the scaling option used:

s1 Creates intervals based on key field 1.

.5 Defines an interval size of .5.

These are the parameters specified:

1 Labels the Down field as key field 1.

/ Computes average cost for each interval. Note that intervals that contain no entries are omitted.

r Rounds cost field to whole numbers.

This is the result of the calculation:

```
*Location. . . .Down. Cause.  Value    .  Cost   . . . .
*========. . . .====.======.==========.=========. . . .
                  0.0                            9
                  0.5                           38
                  1.0                         1270
                  2.0                          550
                  4.0                         7500
```

Example 1Ø: Scaling a Date Key Field

This example demonstrates how to scale data using dates. Following is a portion of the function mask:

```
s1D1(2W/890520/891201/SAT)
*Location.Date  .  Time  . . .  Value   .  Cost   . . .
*========.======.========. . .==========.=========. . .
          1                              /
```

This is the scaling option used:

s1 Creates intervals based on key field 1.

D1 Indicates date scaling in date format 1 (YYMMDD).

2W Creates an interval size of 2 weeks.

890520 Ignores dates before 890520.

891201 Ignores dates after 891201.

SAT Begins each interval on a Saturday.

These are the parameters specified:

1 Labels the Date field as key field 1.

/ Computes average cost for each interval.

Following is the result of the date scaling example:

```
*Location.Date  .  Time  . . .  Value   .  Cost   . . .
*========.======.========. . .==========.=========. . .
          890603                            1455
          890701                              32
          890923                            7500
          891118                              20
```

Note: Note that intervals with no entries are omitted.

Example 11: Scaling a Time Key Field

Following is an example of how to scale data according to time intervals. A portion of the function mask shows use of the option and parameters.

```
s1TO(6H/06:00:00)
*Location.Date  .  Time  .Down. Cause . . .
*========.======.========.====.====== . . .
                 1        +
```

This is the scaling option used:

s1 Scales on key field 1.

TO Indicates times are in format 0 (HH:MM:SS).

6H Creates an interval size of 6 hours.

06:00:00 Ignores times before 06:00:00.

These are the parameters specified:

1 Labels the Time field as key field 1.

+ Computes amount of hours down for each interval.

Following is the result of the time scaling example:

```
*Location.Date  .  Time  .Down. Cause . . .
*========.======.========.====.====== . . .
                 06:00:00 2.40
                 12:00:00 0.50
                 18:00:00 4.70
```

Note: *Note that intervals with no entries are omitted.*

Example 12: Extending Intervals to the Boundaries

Using scaling with the B option, you can include time and date intervals that contain no occurrences within the boundaries you specified. This option is useful to identify intervals where data may be missing or to identify patterns of occurrences.

Following is a portion of a function mask:

```
s1(1/0/5)b
*Location. . . .Down. Cause.  Value   .  Cost   .
*========. . . .====.======.==========.=========.
                1                      +
```

This is the S and B option used:

s1 Creates intervals in the key field 1.

1 Sets the interval size at 1.

0 Begins the first interval at 0.

5 Begins the last interval at 5.

b Extends the intervals to the limits as set by the scaling option (0 and 5 in this example). In this case, even though there may be no data in some of the intervals, the b option includes those intervals in the result.

These are the parameters specified:

1 Specifies the Down field as key field 1.

+ Calculates subtotal for each unique key. In other words, finds the cost per interval.

This is the result of the calculation:

```
*Location. . . .Down. Cause.  Value    .  Cost   .
*========. . . .====.======.==========.=========.
                   0                         124
                   1                        3810
                   2                         550
                   3                           0
                   4                        7500
                   5                           0
```

Note: If the B option had not been included, the intervals that contained no entries (3 and 5) in the input result would have been omitted from the result.

Example 13: Suppressing Headings in Multiple Analyses

The following example demonstrates the use of the H option to suppress headings that would normally appear between multiple analyses. This is a portion of the function mask:

```
h
*Location. . . .Down. Cause.  Value    .  Cost   .Count.
*========. . . .====.======.==========.=========.=====.
 1                                      +
                      1                 /         =
```

First set of parameters:

1	Indicates the key field.
+	Subtotals the cost for each location.

Second set of parameters:

1	Indicates the key field.
/	Calculates the average cost for each system.
=	Counts the number of data lines for each system. In other words, counts the number of entries for each unique key.

Following is the result where the headings between the analyses are omitted:

```
*Location. . . .Down. Cause.  Value      Cost   .Count.
*========. . . .====.======.==========.=========.=====.
 BOSTON                                    1004
 CHICAGO                                  10970
 NEW YORK                                    10
                    CODE1             302.50000     2
                    CODE2             767.80000     5
                    CODE3                  3770     2
```

Example 14: Calculating Total Summarizations

The T option allows you to calculate a total for each computed field. Subtotaling shows a grand total; averaging shows a grand average; entry counting shows total entries; minimum and maximum operators show the smallest or largest values in the field.

Following is a portion of a function mask:

```
t
*Location. . . .Down. Cause.  Value    .  Cost   .Count.
*========. . . .====.======.==========.=========.=====.
 1             +%   =                 >
```

1	Specifies the key field.
+%	Specifies subtotal percentage.
=	Specifies an entry count.
>	Specifies the maximum value.

This is the result of the calculation:

```
*Location. . . .Down. Cause.  Value      Cost   .Count.
*========. . . .====.======.==========.=========.=====.
 BOSTON          19.      4                   900
 CHICAGO         73.      4                  7500
 NEW YORK         9.      1                    10
*               ---- ------             ---------
*               100.      9                  7500
.
. OUTPUT LINES = 3
                    ..... END REPORT .....
```

These calculations show that the Chicago location accounted for 73 percent of the total down time for all locations. There were a total of 9 down-time events. The maximum cost in Boston was $900 and in New York was $10. The maximum cost at any location was in Chicago at $7500. Three locations were included in the report.

Example 15: Displaying Only Specified Result Fields

To include in the result only the fields for which you specified a parameter, use the P option. This is a portion of the function mask used in this example:

```
p
*Location. . . .Down. Cause.  Value     .  Cost   .Count.
*========. . . .====.======.==========.=========.=====.
 1                                    +
```

1 Specifies the key field.

+1 Subtotals the cost for each location.

This is the result of the calculation:

```
*Location.  Cost    .
*========.=========.
 BOSTON        1004
 CHICAGO      10970
 NEW YORK        10
```

Example 16: Reordering Specified Result Fields

In addition to including in the result only the fields for which you specified a parameter, the P option also reorders the fields according to a preset hierarchy:

1. key fields
2. extraction fields
3. computational fields
4. minimum fields
5. maximum fields
6. entry counts
7. percentages

Following is a portion of a function mask that illustrates the reordering capability of the P option:

```
p
*Location. . . .Down. Cause.  Value    .  Cost   .Count.
*========. . . .====.======.==========.=========.=====.
 1              b+   /b     /a         a+
```

1 Specifies the key field.

b+ Labels the Down field as B and subtotals the down time for each key in the key field.

/b Calculates the average value for each unique key in the field labeled B.

/a Calculates the average value for each unique key in the field labeled A.

a+ Labels the Cost field as A and subtotals the cost for each key in the key field.

This is the result of the calculation:

```
*Location.  Cost   .  Value   .Down. Cause
*========.=========.==========.====.======
 BOSTON        1004        251 2.10 0.5250
 CHICAGO      10970 2742.50000 8.20 2.0500
 NEW YORK        10         10    1      1
```

Note: *Note that the fields are reordered according to the hierarchy specified.*

COMBINE

Use the COMBINE run to create an ad hoc report by combining fields from two or more reports. Like the Match (MA) function, the COMBINE run enables you to select fields to match on and include in the receiving report. However, the COMBINE run includes in the report only those fields you select.

Note: *The COMBINE run is part of the Application Power Tools (APT) feature of MAPPER software. For more information about the other APT features, see Appendix F.*

How to Access the Run

Menu Access Procedure

Procedure

1. Type **apt** on the control line and transmit.
2. Select **COMBINE** from the APT menu.
3. Select items from the menus and fill in the subsequent forms. For help in choosing an item or filling in the form, press the **Help** key.

Control Line Procedure

Procedure

Enter the following format on the control line.

Format

COMBINE
Displays the COMBINE Access Method form. For help in choosing a selection, press the **Help** key.

For More Information

For more information about the COMBINE run, enter **apt** on the control line, tab to **COMBINE**, and press **Help**. For more information about the data dictionary and the runs available with Application Power Tools (APT) feature of MAPPER software, see Appendix F.

DATE

Use the Date function to perform operations on dates in reports. These are some of the tasks you can perform with the Date function:

- Compare dates in two fields and put the difference in another field.
- Compare dates in a field to today's date.
- Change the format of a date field, such as from YYMMDD to DD MMM YY. Or change the format of a time field to decimal hours.
- Compute the day of the week.

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Was a report on display?
   Yes: Fill in the function mask.
   No: Fill in the function form then the function mask.

Menu Path

Select Task
  Perform Math Operations
    Process Dates

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

DATE
Was a report on display?
Yes: Displays the function mask.
No: Displays the function form.

DATE *rd*[*c f*]
Displays the function mask.

Format Fields

rdc Report to process.

f Report format in which to display the function mask.

Options

A Processes all line types.

T Converts a time field to decimal hours. The time field must be in the format HH:MM:SS or HHMM. Use the + parameter to specify the field to convert and an = parameter in the field in which to place the converted time. If you do not specify a result field, the converted time is placed into the original time field.

W Converts dates into days of the week. In the field to convert, enter the date format parameter of that field. Enter a different date format parameter and an equal sign in another field so that the date is converted to another format. Enter the : parameter in the field in which to place the 3-letter day of the week.

n Defines the number of days in a workweek, where *n* is 1 to 7. Default = 7.

Arithmetic Parameters

+	Field on which a date operation is performed
-	Field to subtract
=	Field in which to place the result of an operation
:	Field in which to place the day of the week
k	Constant field to add to or subtract from a field
+*n* or -*n*	Constant number to add to or subtract from a field

Date Format Parameters

Symbol	Format	Example
A	YMMDD	00603
B (default)	YYMMDD	900603
C	DD MMM YY	03 JUN 90
D	YDDD	0307
E	YYDDD	90307
F	DDMMYY	030690
G	MM/DD/YY	6/03/90
I	MMDDYY	060390

Outcome

- The function processes the dates and times and displays a result.
- If you compare two dates for which the difference is a negative number, a minus sign precedes the calculated difference.
- If you process a blank date field, asterisks fill the result field.

Guidelines

Dates Used

Make sure all dates processed are between 1944 and 2044.

1100: Dates processed are between 1964 and 2063.

Using Date Formats

Always place the date format before other parameters.

To change the dates in a field to a new format, you must copy them to a different field, defining the new field with the new format parameter.

If you specify a format only one field, all fields are assumed to be in that format. To use two or more date fields in different formats, specify the formats on all fields you use.

Adding and Subtracting a Constant Field

To add or subtract a constant field, follow these steps:

1. Type + or - in the first column of the date field.
2. Type **k** in the field to add or subtract.
3. Type = in the field to contain the computed result.

Adding and Subtracting a Constant Number

To add or subtract a constant number, do the following:

1. Type these values in the field to add to or subtract from:
   - The date format
   - A plus or minus sign
   - The number of days to add or subtract
2. Type = in the field to store the new date.

For example, to add 30 days to a date field in format A, type **a+30** in that field and = in the field to store the new date.

Using Today's Date

To add or subtract today's date, use 0. For example, a+0. See the example "Comparing a Date to Today's Date."

How a Difference in Workdays Is Calculated

When calculating the difference (in workdays) between two dates, and the number of days per week is fewer than seven (specified with the *n* option), the Date function uses the following rules:

- The workweek is assumed to begin on Monday.
- If the starting (earlier) date does not fall on a workday, it is moved backward to the closest workday before calculating the difference.

- If the ending (later) date is not a workday, it is moved forward to the next workday before calculating the difference.
- If both the starting and ending dates are not workdays, both dates are moved forward to the next closest workday before calculating the difference.

IDAT Run

1100: On OS 1100 MAPPER Systems, you can use the IDAT run to generate DATE function masks that you can save, alter, and use again. For information on how to use the iterative runs, see Section 5.

Examples

Example 1: Comparing a Date to Today's Date

In this example, the function compares the Produc Actual date with today's date and places the difference into the Ship Date field. If the date in the Produc Actual field is in the past, a minus sign precedes the calculated difference. If a date is blank, asterisks appear in the Ship Date field.

This is the function mask for this example:

```
.Produc. Ship .
.Actual. Date .
.======.======.
 ******.******.
 +      =
```

+ Identifies the Produc Actual field as the field to compare to today's date.

= Identifies the Ship Date field as the field to store the difference between today's date and the date in the Produc Actual field.

Example 2: Converting a Date Format

In this example, the function converts dates in the Produc Actual field to format I and places them into the Ship Date field. If the Produc Actual field is blank, asterisks appear in the Ship Date field.

This is the function mask for this example:

```
.Produc. Ship .
.Actual. Date .
.======.======.
 ******.******.
 b      i=
```

b Identifies the Produc Actual field, which is in the format YYMMDD, as the field to convert.

i= Identifies the Ship Date field as the field to receive the date from the Produc Actual field, converted to format MMDDYY.

Example 3: Determining the Day of the Week

This example does the following:

- The function converts dates from the Produc Actual field to format I and places them into the Ship Date field.
- The day of the week appears in the Spc Cod field.
- If any dates in the Produc Actual field are blank, asterisks appear in the Ship Date field and blanks appear in the Spc Cod field.

This is the function mask for this example:

```
w
.Produc. Ship .Ship .Spc.
.Actual. Date .Order.Cod.
.======.======.=====.===.
 ******.******.*****.***.
 b       i=           :
```

w Specifies that a date field is to be converted to a day of the week.

b Identifies Produc Actual as the field to convert.

i= Identifies Ship Date as the field to receive the date from the Produc Actual field, converted to format I (MMDDYY). (Note that the = parameter is required with the w option.)

: Identifies Spc Cod as the field to contain the three-letter day of the week (MON, TUE, and so forth).

Example 4: Adding a Constant Number to a Date

In this example, the function adds ten workdays (based on a 5-day workweek) to the Produc Actual field and places the new date into the Ship Date field. If any dates in the Produc Actual field are blank, asterisks appear in the Ship Date field.

This is the function mask for this example:

```
5
.Produc. Ship .
.Actual. Date .
.======.======.
 ******.******.
 b+10   =
```

5 Specifies a 5-day workweek (using the n option).

b+10 Specifies that 10 workdays are to be added to the Produc Actual field.

= Identifies Ship Date as the field to receive the calculated date.

Example 5: Calculating Number of Weeks

In this example, the function compares the dates in the Produc Actual field to today's date and places the difference (in weeks) into the Ship Date field.

This is the function mask for this example:

```
1
.Produc. Ship .
.Actual. Date .
.======.======.
 ******.******.
 +      =
```

1 Specifies a one-day workweek in order to calculate the number of weeks (using the n option).

+ Identifies Produc Actual as the field to compare to today's date.

= Identifies Ship Date as the field in which to place the difference in the number of weeks between the Produc Actual date and today's date.

DECODE (Decode Report)

This section does not apply to the A Series MAPPER System, the BTOS II MAPPER System, or the Personal Computer MAPPER System.

Use the Decode Report (DECODE) function to display report data that was previously encoded with the Encode Report (ENCODE) function. You must use the same encryption key and level used to encode the report. See the Encode function for more information.

1100: Encryption levels are not available on OS 1100 MAPPER Systems.

Prerequisite

The report must be encoded with the ENCODE function.

How to Access the Function

Control Line Procedure

Procedure	Enter the following format on the control line.
Formats	**DECODE** Displays the function form. **1100:** You can also use the following format on OS 1100 MAPPER Systems: **DECODE** ***key*** Displays the function form.
Format Field	***key*** One- to eight-character code that was used to encode the report.

Outcome

The system displays a result containing the restored report data. You can now save or update the data.

System Messages

The system displays a message if one of the following occurs:

- You specified the incorrect key or encryption level.
- The encoded report is corrupted.
- The report is not encoded.

1100: On OS 1100 MAPPER Systems, the decoding process produces invalid characters under those circumstances. Whenever the system produces these invalid characters, it displays a system message indicating an error and flags each invalid character in the result with an SOE character.

If the encoded report is corrupted, characters in the result shown as SOEs cannot be recovered.

Guideline

1100: If you are using a normal ASCII terminal, you may not be able to decode a report that was encoded from a National Character Set (NCS) terminal.

DEL (Delete)

Use the Delete (DEL) function to remove the lines displayed in an update result from the original report and display the updated report.

Prerequisite

An update result from a previous function must be displayed.

How to Access the Function

Menu Procedure

Procedure	Select menu items using the menu path below.
Menu Path	Select Task Modify a Report Delete Data

Control Line Procedure

Procedure	Enter one of the following formats on the control line.
Format	**DEL** [*psw*] Removes the lines displayed in the update result from the original report and displays the updated report.
Format Field	*psw* Password required to perform a delete operation on a write-protected report.

Guidelines

Displayed Lines Deleted

Only the lines displayed in the update result are removed from the original report when you execute the DEL function. If you delete lines from the update result before you execute the DEL function, those lines are not removed from the original report.

Other Functions That Use Update Results

To remove the lines from the original report and retain the lines in the update result, use the Extract (EXT) function. See EXT for more information.

To replace lines in a report with the lines in an update result, use the Update (UPD) function. See UPD for more information.

Functions that Produce Update Results

- CALU (Calculate Update)
- CHG (Change) with the OU option
- LOC (Locate) with the OU option
- MAU (Match Update)
- SU (Search Update)

DELSHR

Use the DELSHR run to remove the link between applications that share specific data dictionary information, making the information unavailable to the applications it was shared with. The DELSHR run does not physically delete an item; it only eliminates the ability of the applications with which it was shared to access it.

Note: The DELSHR run is part of the Application Power Tools (APT) feature of MAPPER software. For more information about the DELSHR run, the data dictionary, and other APT features, see Appendix F.

How to Access the Run

Menu Access Procedure

Procedure

1. Type **apt** on the control line and transmit.
2. Select **DELSHR** from the APT menu.
3. Select items from the subsequent menus. For help in making the selections, press the **Help** key.

Control Line Procedure

Procedure

Enter the following format on the control line.

Format

DELSHR

Displays a menu. For help in making menu selections, press the **Help** key.

For More Information

For more information about the DELSHR run, enter **apt** on the control line, tab to **DELSHR**, and press **Help**. For more information about the data dictionary and the runs available with Application Power Tools (APT) feature of MAPPER software, see Appendix F.

ELT (Element)

This section applies only to the OS 1100 MAPPER System.

Use the Element (ELT) function to copy a MAPPER report to a standard OS 1100 program file or symbolic element, or to a data file.

Prerequisite

The file must be a sector-formatted file with no read or write keys, and it cannot be in read only or write only mode. You must be in the cabinet containing the report you wish to copy or have the report on display.

Note: *Use the Retrieve File (RET [P]) function to copy an OS 1100 file to a MAPPER report. Call your coordinator to see whether you have access to the ELT function.*

How to Access the Function

Control Line Procedure

Procedure	Enter the following format on the control line.
Format	ELT Displays the function form.

Outcome

If the file you are copying is not a currently assigned file, the ELT function assigns it with a maximum granularity of 262,143 tracks.

Guidelines

File Naming Conventions

The qualifier, file name, element, and version must all conform to OS 1100 rules for standard names. Only alphabetic, numeric, and the special characters dollar sign ($) and minus (-) can be included. Please refer to the *ECL Operations and Programming Reference* for complete information on files, elements, and cataloging.

Controlling the Data Copied

When you copy a report to the OS 1100 system, you can control which data is copied. Use the same data control commands you use with the Start function. You can place these commands on any line in the report; the commands take effect from that point in the report. Spaces at the right of data lines are not transferred. Characters that cannot be translated are changed to spaces.

Copying More Than One Report or Drawer

Use the $INCL$ command to copy more than one report or drawer. See Appendix D for more information.

Requesting a New Cycle

You can request a new cycle (+1) for an OS 1100 file. If a new cycle is requested, the file is cataloged public with the same granularity and reserve as the original file.

Saving an Element

If you request to save an element, the file in which you save it must be a program file, not a data file. If you do not enter an element name, the file must be a data file.

Saving an Existing Element or Data File

If you request to save an existing element, the element is marked for deletion and a new element is inserted. If you request to save an existing data file, the file is overwritten.

ELT- (Element Delete)

▽ *This section applies only to the OS 1100 MAPPER System.*

Use the Element Delete (ELT-) function to delete a standard OS 1100 symbolic element from a program file or a data file. The file must be a sector-formatted file with no read or write keys.

You cannot use the ELT- function to delete a MAPPER file.

Prerequisite

Call your coordinator to see whether you have access to the ELT- function.

How to Access the Function

Control Line Procedure

Procedure	Enter the following format on the control line.
Format	ELT- Displays the function form.

ENCODE (Encode Report)

This section does not apply to the A Series MAPPER System, the BTOS II MAPPER System, or the Personal Coumputer MAPPER System.

Use the Encode Report (ENCODE) function to transform report data into unreadable code. The report data cannot be displayed until you supply the correct key using the Decode Report (DECODE) function.

Caution

Remember the encryption key you use. Because the MAPPER system keeps no record of the key, the coordinator cannot tell you the key if you forget it.

How to Access the Function

Control Line Procedure

Procedure Enter the following format on the control line.

Formats **ENCODE**
Displays the function form.

1100: You can also use the following format on OS 1100 MAPPER Systems:
ENCODE *key*
Displays the function form.

Format Field ***key*** One- to eight-character with which to code the report. Valid characters are A through Z and 0 through 9.

Outcome

A result is displayed containing only the heading lines and a message at the top of the screen indicating that the result is protected. (If you specified Y in the **Include Headings?** field, the headings are not displayed.)

You cannot display or update the result until you use the DECODE function.

1100: On OS 1100 MAPPER Systems, a result is displayed with the data scrambled.

Guidelines

- Ask the coordinator whether the ENCODE function is installed on your system.
- Because of high processing overhead, excessive use of this function may affect system performance.
- Do not move encoded data between drawers of different report widths because you will not be able to decode it. Changing the width of the drawer causes all encoded reports in that drawer to be corrupted and therefore unreadable.
- **1100:** Do not update an encoded report. Any change to the encoded report corrupts the report and you will not be able to decode it. Protect your report from updates by using an update password. See the Password (PSW) function for more information.
- You may not be able to decode data with a normal ASCII terminal if the data was encoded with a National Character Set (NCS) terminal. Also, a report containing special NCS characters cannot be encoded with a normal ASCII terminal.
- Encoding a report approximately doubles its size.

ENTRY

Use the ENTRY run to add, change, or delete items in the data dictionary.

Note: *The ENTRY run is part of the Application Power Tools (APT) feature of MAPPER software. For more information about the ENTRY run, the data dictionary, and other APT features, see Appendix F.*

How to Access the Run

Menu Access Procedure

Procedure

1. Type **apt** on the control line and transmit.
2. Select **ENTRY** from the APT menu.
3. Select the desired type of entry from the Data Dictionary Entry menu. For help when choosing the entry type, press the **Help** key.
4. Fill in the form for the specified entry type. For help while filling in the form, press the **Help** key.
5. Type any documentation you would like to include in the data dictionary for the item.
6. Press **Transmlt**, then press **Resume**.

Control Line Procedure

Procedure

1. Enter one of the following formats on the control line.
2. Fill in the form for the specified entry type. For help while filling in the form, press the **Help** key.
3. Type any documentation you would like to include in the data dictionary for the item.
4. Press **Transmit**, then press **Resume**.

Formats

ENTRY
Displays the Data Dictionary Entry menu.

ENTRY,*entry-type*
Displays the form for the specified entry type.

Format Field

entry-type One of the following data dictionary entry types:

F	Field
C	Cabinet, drawer, report
R	Run
E	External subroutine
S	Screen
M	Miscellaneous item
A	Application

Outcome

The ENTRY run updates the data dictionary with the changes you specified.

For More Information

For more information about the ENTRY run, enter **apt** on the control line, tab to **ENTRY**, and press **Help**. For more information about the data dictionary and the runs available with Application Power Tools (APT) feature of MAPPER software, see Appendix F.

EXT (Extract)

Use the Extract (EXT) function to remove the lines displayed in an update result from the original report and redisplay the result as a normal result. You can then do other processing on the extracted lines, such as duplicating the result to save the lines in a permanent report.

Since the new result is no longer an update result, you cannot use it to update the original report again.

Prerequisite

An update result from a previous function must be displayed.

How to Access the Function

Menu Procedure

Procedure	Select menu items using the menu path below.
Menu Path	Select Task Modify a Report Extract Data

Control Line Procedure

Procedure	Enter one of the following formats on the control line.
Format	**EXT** [*psw*] Removes the lines displayed in the update result from the original report and redisplays the result.
Format Field	*psw* Password required to perform an extract operation on a write-protected report.

Guidelines

Displayed Lines Deleted

Only the lines displayed in the update result are removed from the original report when you execute the EXT function. If you delete lines from the update result before you execute the EXT function, those lines are not removed from the original report.

Other Function That Use Update Results

To remove the lines from the original report and discard the update result, use the Delete (DEL) function. See DEL for more information.

To replace lines in a report with the lines in an update result, use the Update (UPD) function. See UPD for more information.

F (Find)

Use the Find (F) function to find character strings in specified fields in a report or drawer and display the report at the first occurrence. (Press **Resume** to see more occurrences.)

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Was a report on display?

   Yes: Fill in the function mask.

   No: Fill in the function form then the function mask. For help while filling in the function form, press **Help.**

Menu Path

Select Task
    Find Data
        Find

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

F

Displays the function form.

F {*rd*[*c*]|*d*[*c*]} [*f*]

Displays the function mask.

Format Fields

rdc Report to process. (Or type - in this field to indicate the current report.)

1100: On OS 1100 MAPPER Systems, you do not need to type the minus (-) to indicate the current report.

dc Drawer to process. Use this field to search all reports in a given drawer.

f Report format in which to search for data (lets you look for data in fields beyond column 80 if you do not already have those columns on display).

Options

A Processes all line types.

C(S) Distinguishes between uppercase and lowercase letters.

1100: C(x) Alters the find process based on the character set order. Ordinarily the system processes the report based on the character set of the drawer. The C option allows you to choose the character set type on which to base the find. Use one of the following:

C(F) Full character set (FCS)
C(L) Limited character set (LCS)
C(S) Strict comparison; distinguishes between uppercase and lowercase letters

R*x*-*y* Searches a range of reports from *x* through *y*.

R*x*,*y* Searches reports *x* and *y*.

R*x*-*y*,*z* Searches reports *x* through *y* and also *z*.

@ Searches for spaces.

/ Searches for the slant (/) character.

Parameters

The character string you want to find is the most common parameter to use with the F function. You can also specify the following parameters:

/ Searches for a slant (/) character when used with the / option. Without the / option, a slant appearing in a parameter line shortens that field (the partial field is processed).

@ Searches for spaces. Use with the @ option: type @ in the columns of the function mask where you want to search for spaces. To find a blank field, type @ in the first column of the field.

R Searches for a range of data.

Outcome

- The system displays the report starting at the line where the data is located. When the search is complete, a system message states that there are no more finds.
- Press **Resume** to continue the search. The system displays the report at the next find that occurs after the current screen.

  Note: *If the string exists on the same screen as a previous find, the F function does not display that line as the next find.*
- If the specified data is not found, a system message is displayed.

Guidelines

Searching for More Than One Character String

To search for multiple strings in the same field, enter each character string on a separate line under the field heading in the mask.

To search for multiple strings in different fields, enter the character strings under different headings on the same line. A find is made when all specified fields contain the requested strings.

To search for multiple strings in different fields and include either one string or the other or both, enter the character strings under different headings and on different lines in the mask. A find is made when either of the specified fields contains the requested strings.

See the "Examples" section for more details.

Searching for a Range of Strings

To search for a range of character strings, follow this example:

1. Type the characters for the lower end of the range in the first line under the mask in the correct field.
2. Type r in the first character position in the next line.
3. On the same line as the R, type the characters for the higher end of the range in the same field as the lower end. Then transmit.

```
*St.Status.
*Cd. Date .
*==.======.
 ** ******
    800101   <-- Lower end of the range.
r   891231   <-- R parameter in column 1 designating a
                 range search; higher end of the range.
```

IFND Run

1100: On OS 1100 MAPPER Systems, you can use the IFND run to generate F function masks that you can save, alter, and use again. For information on how to use the iterative runs, see Section 5.

Examples

Example 1: Searching a Range of Reports

```
r2-10,14
*St.Status.Cust.
*Cd. Date .Code.
*==.======.====.
 ** ****** ****
           intr
           dico
```

`r2-10,14` Search reports 2 through 10 and 14.

`intr`
`dico` Search for these character strings in the Cust Code field.

More Examples

See the S (Search) function for more examples of searching for data. Since the parameters available for the S and F functions are very similar, you can try the examples shown in the S function discussion while you are executing the F function.

Note: *Since no result is created with the F function, some of the S function options are not available with the F function.*

FILE (Create File)

Use the Create File (FILE) function to copy a MAPPER report to a file in either MAPPER format or native data file format.

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Fill in the function form. For help while filling in the function form, press the **Help** key.

Menu Path

Select Task
    Move Data
        Create a Native File

Control Line Procedure

Procedure

Enter the following format on the control line.

Format

FILE

Displays the function form.

Outcome

- If the file does not exist, the function creates the file; if the file already exists, the function overwrites it.
- If you create a file in a format other than MAPPER format, and the original MAPPER report contains tab characters, the function translates tab characters to spaces. See Appendix D for information on translating tab characters to tab or other characters using the $TABA$ command.
- If you create a file with headings, the file keeps information such as the report identifier and the date the report was last updated. The function writes the date line into the file so when you retrieve the file, the date line of the original report is part of the result.

U Series:

- The function redisplays the report you had on display when you requested the function.
- Do not use UNIX edit utilities on files you create in MAPPER format because the results may be unpredictable.
- If your UNIX login does not have read permission for the specified file, you receive this prompt:

```
Your login is not permitted to access this file.

To access the file and continue, provide another login.

Login:
```

A Series:

- An active screen is displayed after the file is created.
- You can create the file in MAPPER format (which is unreadable by CANDE and MARC and contains special control characters) or in standard data file format (which can be processed by CANDE, MARC, and other A Series edit utilities).
- **PC MAPPER:** You can create the file in MAPPER format (which contains control characters and may be unreadable by MS-OS/2$^{\text{TM}}$ text editors) or in standard data file format (which can be processed by OS/2 text editors).

Guideline

U Series: Do not use any UNIX metacharacters when specifying the file name (for example, *, ^, [,], and so on). These characters are interpreted in a special way by the UNIX shell so you may not get the results you are expecting.

A Series: See the *A Series I/O Subsystem Programming Reference* for a description of valid file names.

MS-OS/2 is a trademark of the Microsoft Corporation.

FORMG

Use the FORMG run to create an experimental report. You use a form to specify field order, sizes, edit codes, and report formats. You also have the option of selecting field information from the data dictionary.

Note: The FORMG run is part of the Application Power Tools (APT) feature of MAPPER software. For more information about the FORMG run, the data dictionary, and other APT features, see Appendix F.

How to Access the Run

Menu Access Procedure

Procedure

1. Type **apt** on the control line and transmit.
2. Select **FORMG** from the APT menu.
3. Fill in the FORMG Worksheet form. For help while filling in the forms, press the **Help** key.

Control Line Procedure

Procedure

Enter the following format on the control line.

Format

FORMG

Displays the FORMG Worksheet form. For help while filling in the form, press the **Help** key.

If a report was on display when you typed FORMG, the worksheet includes the field information from the displayed report. If the displayed report was the report 0, the field information includes edit codes.

Guidelines

While filling in the worksheet, you can press **Show** to see what the report will look like. Press the **Return** function key to return to the worksheet.

When the worksheet is complete, press **Create** to place the experimental report on the system as a report. You can then give the report location to the MAPPER system coordinator, who will generate a drawer based on the experimental report.

For More Information

For more information about the FORMG run, enter `apt` on the control line, tab to **FORMG**, and press **Help**. For more information about the data dictionary and the runs available with Application Power Tools (APT) feature of MAPPER software, see Appendix F.

To learn more about creating experimental reports, see the *Manual Functions Training Guide*.

FORMGEN (Form Generation)

This section does not apply to the OS 1100 MAPPER System.

Use the FORMGEN (Form Generation) run to build an experimental report for generating a new drawer. The report is saved in cabinet 0, drawer F. From there, it can be given to the coordinator who uses it to generate a report 0 to define a new drawer.

How to Access the Run

Menu Procedure

Procedure	1. Select menu items using the menu path below. 2. Follow the procedure described in "Building an Experimental Report" in this section.
Menu Path	Select Task Create a Report Use the FORMGEN Run

Control Line Procedure

Procedure	1. Enter the following format on the control line. 2. Follow the procedure described in "Building an Experimental Report" in this section.
Format	**FORMGEN** The Experimental Report Builder table is displayed.

Outcome

An experimental report is built.

Building an Experimental Report

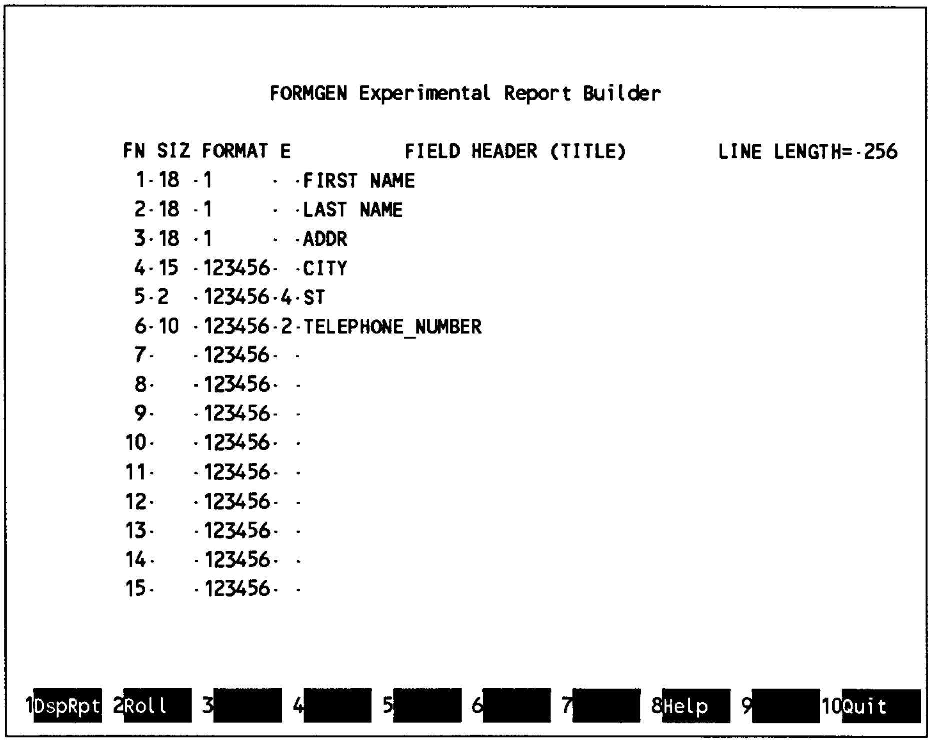
After you access the run through the menu path or the control line, the FORMGEN run displays a vertical list of fields, field sizes, edit codes, and field numbers you can change to modify the form. Follow these guidelines.

Guidelines for Building the Experimental Report

- Fill in the field sizes, field titles, formats, and editing characteristics for the new experimental report.
- To view the experimental report you are building, press **DspRpt**. Then you may save the report, or return and continue building your report.
- The FORMGEN run can design a report with up to 30 fields. If you want to create more than 15 fields, press **RollFw** to define fields 16 through 30.
- The fields are sorted by field number when you display the result.
- You can mark the point at which you wish to divide the title of a field into two lines with the underscore character.
- Field headers may be up to 40 characters. However, if the field size is less than the length of the heading, the title is truncated. To avoid truncation, you can divide the title into two lines with the underscore character.
- You can use the FORMGEN run to rearrange the order in which fields appear. To do this, change the field numbers to the order that you would like them to appear. For instance, to make field 5 the first field, change its field number to 1. To make field 5 the second field, change its field number to 2. When you press **Transmit**, or display the result, the fields are resequenced.

Exercise

1. Enter **formgen**. The Experimental Report Builder is displayed. Fill in the sizes, formats, editing characteristics, and titles for the fields, as shown:

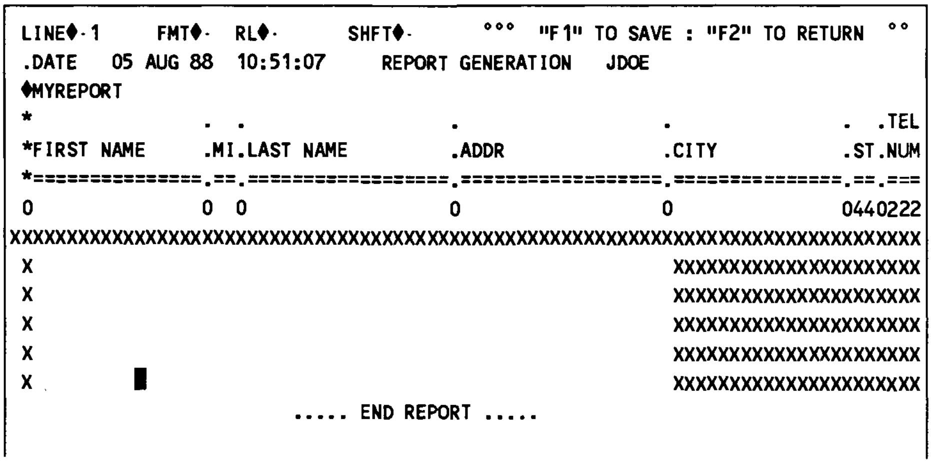


```
                FORMGEN Experimental Report Builder

FN SIZ FORMAT E          FIELD HEADER (TITLE)        LINE LENGTH=·256
 1·18 ·1      · ·FIRST NAME
 2·18 ·1      · ·LAST NAME
 3·18 ·1      · ·ADDR
 4·15 ·123456· ·CITY
 5·2  ·123456·4·ST
 6·10 ·123456·2·TELEPHONE_NUMBER
 7·   ·123456· ·
 8·   ·123456· ·
 9·   ·123456· ·
10·   ·123456· ·
11·   ·123456· ·
12·   ·123456· ·
13·   ·123456· ·
14·   ·123456· ·
15·   ·123456· ·
```

2. Press **DspRpt**.

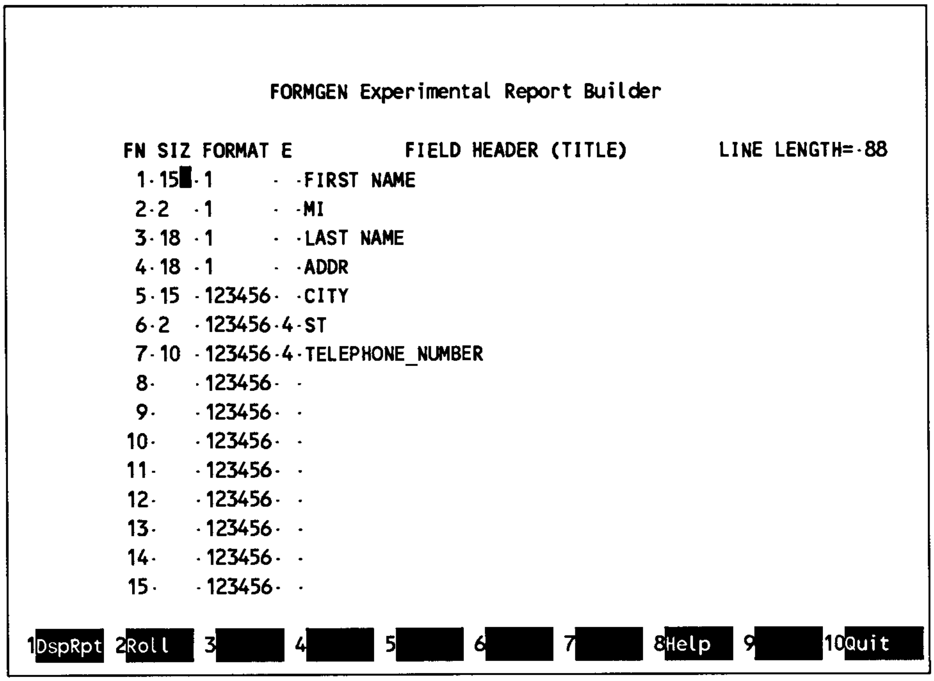
   The displayed result contains the field or headers, format lines, and edit codes specified on the Experimental Report Builder screen.

3. You can save this result by pressing **Save**. Or you can modify the result by changing field headers and inserting additional fields, as illustrated:



```
LINE♦·1     FMT♦·  RL♦·     SHFT♦·        °°°  "F1" TO SAVE : "F2" TO RETURN  °°
.DATE   05 AUG 88  10:51:07     REPORT GENERATION   JDOE
♦MYREPORT
*              .  .                 .                 .              .  .TEL
*FIRST NAME    .MI.LAST NAME        .ADDR             .CITY          .ST.NUM
*==============.==.=================.=================.==============.==.===
0              0  0                 0                 0              0440222
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X                                                      XXXXXXXXXXXXXXXXXXXXX
X                                                      XXXXXXXXXXXXXXXXXXXXX
X                                                      XXXXXXXXXXXXXXXXXXXXX
X                                                      XXXXXXXXXXXXXXXXXXXXX
X         █                                            XXXXXXXXXXXXXXXXXXXXX
                       ..... END REPORT .....
```

4. Press the **Return** function key to return to the Experimental Report Builder. Any changes you made to the result are now reflected in the table.



```
                    FORMGEN Experimental Report Builder

        FN SIZ FORMAT E         FIELD HEADER (TITLE)        LINE LENGTH=·88
         1·15█·1      · ·FIRST NAME
         2·2  ·1      · ·MI
         3·18 ·1      · ·LAST NAME
         4·18 ·1      · ·ADDR
         5·15 ·123456· ·CITY
         6·2  ·123456·4·ST
         7·10 ·123456·4·TELEPHONE_NUMBER
         8·   ·123456· ·
         9·   ·123456· ·
        10·   ·123456· ·
        11·   ·123456· ·
        12·   ·123456· ·
        13·   ·123456· ·
        14·   ·123456· ·
        15·   ·123456· ·

 1DspRpt 2Roll   3       4       5       6       7       8Help   9       10Quit
```

Note that **LINE LENGTH** now reads 88. The table keeps track of the length of the report as you increase the length. This number reflects any change you make that increases the length beyond 80 columns.

If you later decrease the length, the **LINE LENGTH** number does not automatically decrease. To recalculate the length, tab to **LINE LENGTH** and change the number to 80. Then place the cursor below the last field you have changed and press **Transmit**.

You can also arbitrarily set the length to more than what is shown (up to 256 columns). If you try to change the length to below the calculated length, the change is ignored.

5. To save the result, press **DspRpt**. Then press **Save**.

I (Index)

Use the Index (I) function to create a listing of a specified number of lines from all reports in a drawer, starting at the date line of each report.

How to Access the Function

Menu Procedure

Procedure	Select menu items using the menu path below.
Menu Path	Select Task Find Data Display Drawer Index

Control Line Procedure

Procedure	Enter one of the following formats on the control line.	
Formats	**I**	Displays the function form.
	I [*q*]*d*	Displays a listing of the specified number of lines from all reports in the selected drawer.
Format Fields	*q*	Number of lines to display from each report. Default = heading lines (up to eight) from each report.
	d	Drawer to be indexed.

Outcome

A result is displayed containing the following lines:

- The date line for the result.
- A line containing the number of reports in the drawer and the total number of lines in the drawer.
- The specified number of lines from each report starting with the date line in each report. The right side of each date line contains the number of lines in the corresponding report.

  Note: *In the case of a limited-access report, such as a read-protected or encoded report, only the date line and the title line are displayed. The contents of the report is not displayed on the screen.*

Guidelines

Specifiying the Number of Lines

To see only the heading lines from each report, leave the *q* field blank in the function form or format.

To see all of the lines in each report, enter a number larger than the size of any report in the *q* field.

Note: *There is no maximum number for the* q *field, however, since a request to index many lines from each report can affect system performance, use caution when indexing large drawers.*

Control Line Not Included

When determining the number of lines to include from each report, remember that the control line is not considered part of a report.

ICAL (Iterative Calculate)

Use the ICAL (Iterative Calculate) run to create equation sets that you can save, alter, and use again. With the ICAL run, you can use any functionality available with the Calculate (CAL) function, but with greater ease. For example, you can do the following tasks:

- Save equations and options in the function mask and use them again.
- Request existing equation sets and alter them without repeatedly filling in entire masks.
- Save equations in the form of a run statement.

Prerequisite

A report must be on display.

How to Access the Run

Menu Access Procedure

Procedure	1. Select menu items using the menu path below. 2. Fill in the function form then the function mask.
Menu Path	Select Task Perform Math Operations Use the ICAL Run

Control Line Procedure

Procedure	Enter one of the following formats on the control line.
Formats	**ICAL[*]** Displays the function form. **ICAL[* *eqnm,r*]** Displays the function mask.

continued

Control Line Procedure (cont.)

Format Fields

* — Symbol used to specify that field names are to be used in place of column character positions when a run statement is produced.

eqnm — The name of an existing equation set, a new equation set, or blank. (If you leave it blank, you name the new equation set when you save it.) The name can be 1 to 12 characters; it must begin with a letter and contain no spaces.

r — Report containing predefined CAL functions. Specify the report when equations are in a report other than the data report on display.

Note that if you leave the *eqnm* field blank, you cannot specify a report in the *r* field.

Options

Use the same options as those used for the Calculate function.

Parameters

Use the same parameters as those used for the Calculate function.

Outcome

Calculations Performed

The ICAL run calculates equations just as the CAL function does.

Saved Equation Sets

If you saved an equation set, the ICAL run places it in the current report (or the specified report with that drawer) along with its name. The equation set appears as a set of asterisk lines at the end of the report. (An equation set is the set of options, headings, and parameters from the completed mask.)

If you saved an equation set, the ICAL run actually updates the report; it does not create a result.

If a result was on display when you started the ICAL run, the run places the equations at the end of the result. (To save the equation sets, remember to replace or duplicate the result, creating a permanent report.)

After Calculating

After calculating, you can use the keys displayed on the function key bar, as described in "ICAL Function Key Bar" in this section.

Guidelines

- You cannot save equations until after you have performed calculations with them by transmitting from the function mask. You can save several equation sets in the same report, each with a different name.
- You can view the three most recent equation sets by pressing the **Cycle** key three times. Pressing it a fourth time displays the most recent equation set.
- You can include as many equations in each set as will fit on six lines.

ICAL Function Key Bar

Mask Redisplays the function mask, allowing you to change the equations and options and recalculate.

Re-Dsp Redisplays the original report or result.

Cycle Cycles back through the three most recent masks. (See Guidelines for details.)

Save Displays a menu in which you can do one of the following:

- Type the name of the equation set to be saved and then transmit. (If you requested an existing equation set by the name you specify in this menu, that equation set is replaced by the current equation set.)
- Press **Resume** to display the most recently processed mask.
- Press **DspRun** to display the run statement created from the most recently processed mask.

Exercise

To help you understand the ICAL run, try this exercise involving calculating, changing options and equations, and saving equations:

Calculate

1. Duplicate report 1C0 and display the duplicate report.
2. Using either the menu procedure or the control line procedure, start the ICAL run.
3. Enter some options and equations. For ideas, see the Calculate (CAL) function.
4. Transmit to see the computed result.

Change Options or Equations

1. Press the **Mask** key to receive the completed mask and change an option or equation.
2. Transmit to receive the newly calculated result.

Save Equations

1. Press the **Save** key to display the Save Equation Set menu.
2. Enter an equation set name in the menu and transmit.
3. Press the **Re-Dsp** key and roll to the end of the report to see the saved equation set.

INFO

Use the INFO run to display a result containing the data dictionary information for one or more items.

Note: *The INFO run is part of the Application Power Tools (APT) feature of MAPPER software. For more information about the INFO run, the data dictionary, and other APT features, see Appendix F.*

Prerequisite

The information for the items you request must already exist in the data dictionary.

How to Access the Run

Menu Access Procedure

Procedure	1. Type **apt** on the control line and transmit. 2. Select **INFO** from the APT menu. 3. Fill in the run form. For help while filling in the form, press the **Help** key.

Control Line Procedure

Procedure Enter one of the following formats on the control line.

Formats

INFO

Was a report on display?

Yes: Displays the data dictionary information for the displayed report and all its fields.

No: Displays the run form. For help while filling in the form, press the **Help** key.

INFO,*id,entry-type,appnm*

Displays the data dictionary information for the item requested.

Format Fields

id The identifier name of the dictionary item or its alias.

entry-type One of the following data dictionary entry types:

F	Field
C	Cabinet, drawer, report
R	Run
E	External subroutine
S	Screen
M	Miscellaneous item
A	Application

appnm The name of the application.

Outcome

The INFO run displays a result containing a representation of the data dictionary information you requested.

For More Information

For more information about the INFO run, enter **apt** on the control line, tab to **INFO**, and press **Help**. For more information about the data dictionary and the runs available with Application Power Tools (APT) feature of MAPPER software, see Appendix F.

IU (Index User)

Use the Index User (IU) function to create a listing of a specified number of lines from selected reports in a drawer.

You can choose any combination of these parameters:

- Reports created or updated by a specific user or by any user.
- Reports with specific last update start dates, end dates, or both.
- Range of reports.

How to Access the Function

Menu Procedure

Procedure	Select menu items using the menu path below.
Menu Path	Select Task Find Data Display Drawer Index by User

Control Line Procedure

Procedure	Enter one of the following formats on the control line.	
Formats	**IU** Displays the function form.	
	IU [*q*]*d* Displays a result containing lines from the specified reports.	
Format Fields	*q*	Number of lines to display from each report. Default = heading lines (up to eight) from each report.
	d	Drawer to index.

Outcome

A result is displayed containing the following lines:

- The date line for the result.
- Lines containing the number of reports that meet the selection criteria and the user-id requested for the index.
- Lines containing headings that identify each field, such as USER and UPDATE DATE, and a heading divider line.
- The specified number of lines from each report starting with the date line in each report. The right side of each date line contains the number of lines in the corresponding report.

  Note: *In the case of a limited-access report, such as a read-protected or encoded report, only the date line and the title line are displayed. The contents of the report is not displayed on the screen.*

Guidelines

Specifying the Number of Lines

To see only the heading lines from each report, leave the *q* field blank in the function form or format.

To see all of the lines in each report, enter a number larger than the size of any report in the *q* field.

Note: *There is no maximum number for the* q *field, but since a request to index many lines from each report can affect system performance, use caution when indexing large drawers.*

Control Line Not Included

When determining the number of lines to include from each report, remember that the control line is not considered part of a report.

KILL

Use the Kill function to terminate an active remote run or background run.

Prerequisite

You must be the person who started the run.

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Fill in the function form. For help while filling in the function form, press the **Help** key.

Menu Path

Select Task
    Use Utilities
        Terminate a Run

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

KILL
    Displays the function form.

KILL {*run*|,*sn*|*run,sn*}
    Terminates the run and returns a message confirming the termination.

Format Fields

run Name of the run to terminate.

sn Number of the station where the run is executing.

Guidelines

Locating the Run and Station

Use the System function to locate the run name and station number of the run you want to terminate.

Terminating Runs

Terminate runs in one of these ways:

- Enter the run name alone to terminate all background runs with that name that you started on any station.
- Enter the station number alone to terminate the background run that you started at that station.
- Enter both the run name and station number to terminate a specific run that you started at the specified station.

Queued Background Runs

A queued background run can be terminated only by the MAPPER system coordinator.

LISTS

Use the LISTS run to display lists of specific information from the data dictionary, including:

- All applications within your department
- Specific types of entries, such as all fields or all runs
- All runs and external subroutines that access a specific field
- All reports that access a specific field

Note: The LISTS run is part of the Application Power Tools (APT) feature of MAPPER software. For more information about the LISTS run, the data dictionary, and other APT features, see Appendix F.

How to Access the Run

Menu Access Procedure

Procedure

1. Type **apt** on the control line and transmit.
2. Select **LISTS** from the APT menu.
3. Select items from the menus and fill in the subsequent forms. For help in choosing an item or filling in the form, press the **Help** key.

Control Line Procedure

Procedure

Enter the following format on the control line.

Format

LISTS

Displays the LISTS menu. For help in choosing a selection, press the **Help** key.

For More Information

For more information about the LISTS run, enter `apt` on the control line, tab to **LISTS**, and press **Help**. For more information about the data dictionary and the runs available with Application Power Tools (APT) feature of MAPPER software, see Appendix F.

LOC (Locate)

Use the Locate (LOC) function to find a character string anywhere within one report and display the report at the first occurrence. (Press **Resume** to see more occurrences.)

To locate strings quickly, use the following format in the control line:

LOC *tgtstr*

where *tgtstr* is the string of characters to locate.

How to Access the Function

Menu Procedure

Procedure	1. Select menu items using the menu path below. 2. Was a report on display? Yes: Fill in the function mask. No: Fill in the function form then the function mask. For help while filling in the function form, press **Help**.
Menu Path	Select Task Find Data Locate

Control Line Procedure

Procedure	Enter one of the following formats on the control line.
Formats	**LOC *tgtstr*** or **LOC** [*rdc f*];/*tgtstr*/[*o*] Displays the first occurrence of the specified target string within the current report or specified report. **LOC** Report on display? Yes: Displays the function mask. No: Displays the function form.

continued

Control Line Procedure (cont.)

Formats (cont.)

LOC ***rd*[*c f*]**

Displays the function mask.

Format Fields

rdc Report to process.

f Report format in which to search for data (lets you look for data in fields beyond column 80 if you do not already have those columns on display).

; Required character.

/ Delimiter identifying the character string boundaries. Can be any character except those used in the target string or a caret (^).

tgtstr Character string to be located.

o Options. In this function format, the A, F, and M options are always assumed.

Options

A Processes all line types, ignoring the character in column 1 of the function mask. For example, if the target string is |abc (| represents a tab character), the A option looks for the string abc on all line types.

When you specify the target string using the control line procedure, the A, F, and M options are always assumed.

B[*n*] Includes *n* (1 to 9) lines above the data match when the report is displayed. Do not use this option with the O option. Default = 1.

C Distinguishes between uppercase and lowercase letters.

1100: On OS 1100 MAPPER Systems, this option applies only to full character set (FCS) reports.

continued

Options (cont.)

F Processes all line types and includes in the target the character in column 1 of the function mask.

When you specify the target string using the control line procedure, the A, F, and M options are always assumed.

M Designates the first character of the target string as the line type designator. Use with the F option to find the string on all line types, even when the string starts in column 1. You can also use the M option in runs to specify the line type to process. For example, if the target string is |abc (| represents a tab character), the M option looks for the string abc on tab lines.

When you specify the target string using the control line procedure, the A, F, and M options are always assumed.

O Creates a result containing each line on which the character string occurs.

OU Creates an update result containing each line on which the character string occurs. This option can be used only with reports, not with results.

S Starts the search at the first line on the screen.

S*x* Starts the search at line *x*, where *x* is a positive number.

S*x*-*y* Searches lines *x* through *y*.

S*x*,*n* Starts the search at line *x* and searches *n* lines.

T*x* Specifies any character as a wildcard character. Default wildcard character = blank. See the examples in this section. Do not use the wildcard character as the first character in the character string.

U Resumes the search beyond the lines on display, instead of on the next line.

Outcome

- The system displays the report starting at the first data line containing the specified string. If you use the O or OU option, a result or update result is displayed.
- If the specified data is not found, a system message is displayed.

Guidelines

Continuing the Search

To continue the search, press **Resume**.

Completing the Function Mask

To fill in the function mask, follow these steps:

1. On the first line under the mask, type the line type to process in the first column, followed by the target string.
2. Press the **Erase to End of Line** key.
3. Move the cursor to the next line and transmit.

Finding a String in a Specific Field

To find a string in a specific field, erase all asterisks in the function mask except those in the field you want to search.

Finding a String in Freeform Lines

The fastest way to find a string in freeform lines is to use the following format:

```
LOC tgtstr
```

Examples

Example 1

Locating the character string arco in the current report:

```
loc arco
```

Example 2

Locating the character string green in report 2B0 using the S option to begin at line 30:

```
loc 2b0;/green/s30
```

Example 3

Creating a result containing all occurrences of the character string 90/6, using the character - as the target string delimiter:

```
loc ;-90/6-o
```

Example 4

This example locates the string r m (the letter R, two spaces, and the letter M) in report 2B0. Because the spaces in the character string are wildcards, possible finds might be room, ream, roam, and so on.

```
loc 2b0;/r  m/
```

Example 5

This example locates all dates in DATE1$ format (YYMMDD) where the year 1990 is filled in along with any month number, but no day (where there are spaces in the DD portion of the date). The T option changes the wildcard character to a dollar sign ($) so that the request can find spaces.

```
loc ;/90$$  /t$
```

MA (Match)

Use the Match (MA) function to compare the data in one or more fields of two reports and, optionally, copy data from an issuing report to a receiving report, creating a result.

Prerequisite

The receiving report must be displayed. Use the I option if the issuing report is displayed.

How to Access the Function

Menu Procedure

Procedure	1. Select menu items using the menu path below. 2. Was a report on display? Yes: Fill in the function mask. No: Fill in the function form then the function mask.
Menu Path	Select Task Compare Reports Match Data between Two Reports

Control Line Procedure

Procedure	Enter one of the following formats on the control line.	
Formats	**MA** Displays the function form.	
	MA *rd*[c *f*] Displays the function mask.	
Format Fields	*rdc*	Issuing report or, if you use the I option, the receiving report.
	f	Report format in which to match data (lets you match data in fields beyond column 80 in the second report).

Options

A Matches all line types.

B Blends issuing and receiving report lines in a result. Data must be presorted. Do not use this option with the M or N options or with the Match Update (MAU) function.

C(S) Distinguishes between uppercase and lowercase letters.

D Deletes match information lines from the result.

E Does not move blank fields from the issuing report. For matched items appearing in multiple lines of the issuing report, if the last item of the group is blank, the function does not move the data.

F Does not fill move fields in a no-match condition. The F option is ignored with the MAU function.

I Specifies that an issuing report instead of receiving report is displayed.

M Includes in the result only lines containing fields that match between the compared reports. Do not use this option with the B option. The MAU function assumes the M option.

N Includes in the result only lines containing fields that do not match between the compared reports. Do not use this option with the B option.

P Specifies that issuing and receiving reports are presorted by fields to be matched. Do not use this option with the Q option. See Guidelines for more information.

Q Specifies quick execution of match data by assuming the data is presorted. When you specify the Q option, you guarantee that the reports are sorted, so the system does not check for unsorted fields. Do not use this option with the P option. See Guidelines for more information.

S Includes in the result only lines containing fields that match, appearing in the same order as the issuing report.

Parameters

Since the length of the fields must be identical, adjust the field size by erasing or adding asterisks in the function mask.

1 - 5 Specifies the sequence of fields to match, beginning with 1. Type each parameter in the first column of the field.

A - M Specifies the sequence of fields to copy, beginning with A. Type each parameter in the first column of the field.

Outcome

- The function compares data in fields of the receiving report to the corresponding fields in the issuing report and creates a result containing the same headings as, and with the data lines in the same order as, the receiving report.
- If you specified data to be copied, the copied data is displayed in the result. Whenever a matched item appears in multiple lines of the issuing report, the data copied to the receiving report field is from the last matched line of the group.
- During the match process, the system locks both reports to prevent changes by other users.
- The following information is displayed at the top of the result:
  - The number of matches or nonmatches (depending on the options you selected).
  - The number of lines compared.

  Use the D option to omit this information from the result.

Guidelines

Presorting Data

Use the P option to presort fields to match in the issuing and receiving reports. Presorting and using the P option increases processing efficiency.

You must presort the data if you are matching two different line types.

Flagging No-Match Occurrences

To flag fields where no matches occur, enter characters such as asterisks or zeros after the alphabetic parameters in the receiving mask.

Compare to Search and Sort

The Search (S) and Sort functions are similar to the MA function. Refer to those functions to be certain you are using the best function.

Match Update Available

To create an update result that allows you to blend changed lines back into the original report or delete lines from the original report, use the MAU function. See the online help system (HELP,MAU) for more information.

Reports Wider Than 8Ø Characters

The displayed report can be in any format, including a temporary format selected with the Create Temporary Format (VIEW) function, but the only way to request a function mask containing columns beyond column 80 in the other report is to specify a standard format (defined in report 0) in the function form.

If you must match or copy data in the issuing report beyond column 80, either display the columns you need using the VIEW function and use the I option (issuing report on display), or use a format containing the columns you need so that you can request it in the function form.

If you need more formats defined in report 0, see your MAPPER system coordinator.

Examples

Example 1: Matching and Moving Items in Fields

This example copies a field when items in two other fields match. Following is the function mask for this example. Note that the issuing report (1D0) is on top and the receiving report (2B0) is on the bottom.

```
* Product .       .Cust.      .Sale.
*  Type   .       .Code.      .Rep .
*=========. . . . .====. . . . .====.
 *********         ****        ***
 1                 2           a

* Product .       .Cust.      .Spc.
*  Type   .       .Code.      .Cod.
*=========. . . . .====. . . . .===.
 *********         ****        ***
 1                 2           a
```

1, 2 Identify the fields to match in each report.

a Identifies the field to copy and where it is to be placed in the receiving report whenever items in both the Product Type and Cust Code fields match.

Here is the partial result of this example (dots indicate the portions of the result not shown here):

```
* Product .      .Cust.      .Spc.
*  Type   .      .Code.      .Cod.
*=========. . . . .====. . . . .===.
 BLACKBOX1        AMCO
 BLACKBOX1        AMCO
 BLACKBOX2        INTR
 BLACKBOX4        ARCO        LSJ
 BLACKBOX5        FEDS
 BLACKBOX5        AMCO
 BLACKBOX0        FEDS        PLR
                .
                .
                .
 BLACKBOX7        AMCO
 BLACKBOX7        INTR
 BLACKBOX7        FEDS
 BLACKBOX8        FEDS        PLR
 BLACKBOX8        FEDS        PLR
```

Example 2: Flagging a No-Match Occurrence

This example copies a field when items in the Product Type fields match, and fills the field with asterisks whenever the items in the Product Type fields do not match. Following is the function mask for this example. Note that the issuing report (1D0) is on top and the receiving report (2B0) is on the bottom.

```
* Product .       .Sale.
*  Type   .       .Rep .
*=========. . . . .====.
 *********         ***
 1                 a

* Product .       .Spc.
*  Type   .       .Cod.
*=========. . . . .===.
 *********         ***
 1                 a***
```

1 Identifies the field to match in each report.

a Identifies the field to copy whenever items in the Product Type fields match.

a*** Indicates that the receiving field is to be filled with asterisks whenever the items in the Product Type fields do not match.

This is the partial result of this example (dots indicate the portions of the result not shown here):

```
* Product .       .Spc.
*  Type   .       .Cod.
*=========. . . . .===.
BLACKBOX1         ***
BLACKBOX1         ***
BLACKBOX2         ***
BLACKBOX4         LTR
BLACKBOX5         ***
BLACKBOX5         ***
BLACKBOX0         PLR
.
.
.
BLACKBOX7         ***
BLACKBOX7         ***
BLACKBOX7         ***
BLACKBOX8         PLR
BLACKBOX8         PLR
```

MSG (Message Waiting)

Use the Message Waiting (MSG) function to display a message sent to your station or user-id.

Prerequisite

Your terminal must be beeping, with **msg** displayed in the upper right corner of your screen.

1100: On OS 1100 MAPPER Systems, **MsgW** is displayed in the lower right corner of your screen.

How to Access the Function

Press the **Message** key or use one of these procedures to view a message.

Menu Procedure

Procedure	Select menu items using the menu path below.
Menu Path	Select Task Send and Receive Receive a Message

Control Line Procedure

Procedure	Enter the following format on the control line.
Format	**MSG**

Outcome

The MSG function displays the message as a result in the cabinet and drawer it was sent from.

For remote sites, the result resides in cabinet 0, drawer F. If you duplicate the result, the duplicate report resides in cabinet 0, drawer F.

1100: On OS 1100 MAPPER Systems, you cannot send messages to a remote site.

Guidelines

Viewing Consecutive Messages

To display the message following the one displayed, perform the MSG function again.

Deleting Messages

To delete a message from the queue, type `ok` on the control line of the displayed message and transmit, then press the **Return** function key. See the Acknowledge Message (OK) function in the online help system (HELP,OK) for more information.

OS (Operating System Interface)

This section applies only to the Personal Computer MAPPER System.

Use the Operating System Interface (OS) function to interface with the operating system and allow the execution of native PC operating system commands.

Prerequisite

This statement should be used only by experienced OS/2 users.

How to Access the Function

Menu Procedure

Procedure	Select menu items using the menu path below.
Menu Path	Select Task Use Utilities Access Operating System

Control Line Procedure

Procedure	Enter one of the following formats on the control line.	
Format	**OS *o* [*cmd*]**	
Format Fields	*o*	Options field. See Options. Enter options in lowercase characters.
	cmd	Command to be performed. Standard path searching is done to find the command.

Options	
-b	Executes the command in the background and is minimized as an icon on the screen.
-f	Causes the output of the command to be returned as a result in drawer A. If you enter a file name immediately following the -f option, the statement places a copy of the output from the command into that file.

Guidelines

Using the -b Option

If you use the -b option but do not specify a command, the operating system command interpreter (CMD.EXE) session is displayed as an icon. You must open the icon, execute any commands, and exit back to your MAPPER session. Your MAPPER session shows a WAIT message until you exit from the CMD.EXE session.

Example

This statement accesses the native OS/2 operating system, obtains a directory listing of the \mapper directory on drive C:, and places the output in the file c:\mydir and in a drawer A result:

```
os -fc:\mydir dir c:\mapper
```

PASS

Use the PASS run to share data dictionary information with users of different applications or to share it with the public dictionary. With the PASS run, you can share information with other applications and keep track of who is using what information.

You can share all data dictionary items from your application or you can specify which items to share. It is important to note that the PASS run shares only data dictionary information; it does not share the actual data.

Note: *The PASS run is part of the Application Power Tools (APT) feature of MAPPER software. For more information about the PASS run, the data dictionary, and other APT features, see Appendix F.*

How to Access the Run

Menu Access Procedure

Procedure	1. Type **apt** on the control line and transmit. 2. Select **PASS** from the APT menu. 3. Fill in the run form. For help while filling in the forms, press the **Help** key. 4. Select items to be passed using the PASS Item Selection menu. 5. Press **Resume**.

Control Line Procedure

Procedure	Enter the following format on the control line.
Format	**PASS** Displays the run form. For help while filling in the form, press the **Help** key.

For More Information

For more information about the PASS run, enter apt on the control line, tab to **PASS**, and press **Help**. For more information about the data dictionary and the runs available with Application Power Tools (APT) feature of MAPPER software, see Appendix F.

PR (Print)

Use the Print (PR) function to print a report on a system printer. For the system printer, use a valid device name. Check with the MAPPER system coordinator.

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Fill in the function form. For help while filling in the function form, press the **Help** key.

Menu Path

Select Task
  Print
    Print on a System Printer

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

PR

Displays the function form.

PR *rd*[c *f*]

Places the report in a queue for the requested printer.

1100: This format is not available on OS 1100 MAPPER Systems.

Format Fields

rdc Report to print.

f Format of the report to print: 0 through 25. Default = basic, unshifted format.

Outcome

The function redisplays the report on your screen and places it in a queue for the requested printer.

Inserting Page Breaks

To move the printer paper to the top of the next page, type the .EJECT command in the report where you want the page break. You can substitute the period with any character except a space. The command must begin in column 1.

Using Predefined Print Forms

This section on using predefined print forms applies only to the OS 1100 MAPPER System.

You can use the PRINTFORM run to create, edit, and delete banner reports and predefined printer forms.

For information about using this run, and for information on banners and predefined forms, enter **printform,help**.

Format

```
PRINTFORM
```

System Printer Control Commands

This section on system printer control commands applies only to the OS 1100 MAPPER System.

The following is a list of commands you can type in a report that controls printing on system printers. A command must begin in column 1, and it can be typed in uppercase or lowercase letters.

Command	Description
.EJECT	Go to the top of the next page. The period can be any character except a blank.
$HOME$	Go to the top of the next page (same as .EJECT).
$MARG$ *lpp,tm, bm,lpi,lsp*	Set the page parameters: *lpp* Lines per page (1-192) *tm* Top margin (0-99) *bm* Bottom margin (0-99) *lpi* Lines per inch (6 or 8) *lsp* Line spacing (1-3) The parameters are optional. If you leave out one of the parameters, its current value remains in effect. Type an asterisk instead of a value if you want to restore the system default value.
$HDNG$ *o,pgn, ejct?,pghdtxt*	Control page heading: *o* Blank = print normal page headings, N = do not print page headings, and X = do not print page numbers or dates in the page headings. *pgn* (Optional) The starting page number for the next printed page. Change the page number. *ejct?* Y = position the paper at the top. *pghdtxt* Up to 96 characters of data can be used in the text of the page heading.
$SKIP$ *n*	Skip *n* lines in the report being printed.

continued

continued

Command	Description
$SKIP$ *x*	Begin skipping report lines until an $ESKP$ image is encountered that has the matching termination character *x*, where *x* is any nonnumeric character.
$ESKP$ *x*	End skipping lines begun by $SKIP$ *x*, where *x* matches the nonnumeric character in $SKIP$ *x*. To end skipping lines in any case, type $ESKP$ without a nonnumeric character following it.
$SKPL$ *n*	Skip *n* lines on this page, where *n* is a number. If *n* is greater than the number of lines remaining on the page, go to the top of the next page.
$BLOK$ *x*	Block-print *x*, where *x* is from one to eight alphanumeric characters and hyphens and colons. (For example, NEWUSERS.) If the block does not fit on this page, go to the top of the next page before printing it.
$USER$	Block-print the active user-id (up to eight characters).
$DATE$	Block-print the current date (MM/DD/YY).
$TIME$	Block-print the current time (HH:MM:SS).
$DEPN$	Block-print the user's department number (DEPT *nnn*).
$DEPT$	Block-print the user's department name (up to eight characters).

PULL

Use the PULL run to access information in the public dictionary.

Note: The PULL run is part of the Application Power Tools (APT) feature of MAPPER software. For more information about the PULL run, the data dictionary, and other APT features, see Appendix F.

How to Access the Run

Menu Access Procedure

Procedure

1. Type **apt** on the control line and transmit.
2. Select **PULL** from the APT menu.
3. Fill in the run form. For help while filling in the forms, press the **Help** key.
4. Select items to be accessed using the PULL Item Selection menu.
5. Press **Resume**.

Control Line Procedure

Procedure Enter the following format on the control line.

Format **PULL**

Displays the run form. For help while filling in the form, press the **Help** key.

For More Information

For more information about the PULL run, enter **apt** on the control line, tab to **PULL**, and press **Help**. For more information about the data dictionary and the runs available with Application Power Tools (APT) feature of MAPPER software, see Appendix F.

RET (Retrieve File)

This section does not apply to the OS 1100 MAPPER System. See "RET [P] (Retrieve File): OS 1100" in this section for a description of the RET function as it applies to the OS 1100 MAPPER System.

Use the Retrieve File (RET) function to retrieve a native data file into the MAPPER database as a result.

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Fill in the function form. For help while filling in the function form, press the **Help** key.

Menu Path

```
Select Task
    Move Data
        Retrieve a Native File
```

Control Line Procedure

Procedure

Enter the following format on the control line.

Format

RET

Displays the function form.

Outcome

- If a file was originally created with headings and you retrieve it with headings, the result contains two sets of headings.
- If a file was created with no headings and you retrieve it with no headings, the result contains no headings, only the date line.
- If you retrieve a file in MAPPER format that is wider than the drawer into which it is being retrieved, the function truncates (drops) any extra characters at the end of each line. If the file is not in MAPPER format, extra characters wrap to the next line.

- The function always adds a date line to a file that is not in MAPPER format. If you create file as a native data file with headings and retrieve it without headings, the result contains two date lines.

U Series:

- If your UNIX login does not have read permission for the specified file, you receive this prompt:

```
Your login is not permitted to access this file.

To access the file and continue, provide another login.

Login:
```

RET [P] (Retrieve File): OS 1100

This section applies only to the OS 1100 MAPPER System. See "RET (Retrieve File)" in this section for a description of the RET function as it applies to other MAPPER systems.

Use the Retrieve File (RET [P]) function to retrieve OS 1100 program files, symbolic elements, or data files.

How to Access the Function

Menu Procedure

Procedure	1. Select menu items using the menu path below. 2. Fill in the function form. For help while filling in the function form, press the **Help** key.
Menu Path	Select Task Move Data Retrieve a Native File

Control Line Procedure

Procedure	Enter one of the following formats on the control line.
Formats	**RET P** Displays the function form. **RET** (to retrieve a file with the same qualifier as MAPPER) Displays the function form.

Outcome

Displays the retrieved file as a result.

RETR (Retrieve Report)

▽ *This section applies only to the OS 1100 MAPPER System.*

Use the Retrieve Report (RETR) function to retrieve a version of a MAPPER report that existed at the time of the last purge or merge process. It is especially useful if you accidentally delete a report.

If you are unable to retrieve the version of the report that you need, call your coordinator, who might be able to retrieve an earlier version of the report for you.

How to Access the Function

Menu Procedure

Procedure	1. Select menu items using the menu path below. 2. Fill in the function form. For help while filling in the function form, press the **Help** key.
Menu Path	Select Task Use Utilities Retrieve a Report from History

Control Line Procedure

Procedure	Enter the following format on the control line.
Format	**RETR** Display the function form.

Outcome

Displays the retrieved report as a result.

RS (Run Status)

This section applies only to the OS 1100 MAPPER System.

Use the Run Status (RS) function to display the status of all runs (including background batch, and remote runs) started with your user-id.

How to Access the Function

Menu Procedure

Procedure	Select menu items using the menu path below.
Menu Path	Select Task Use Utilities Check Run Status

Control Line Procedure

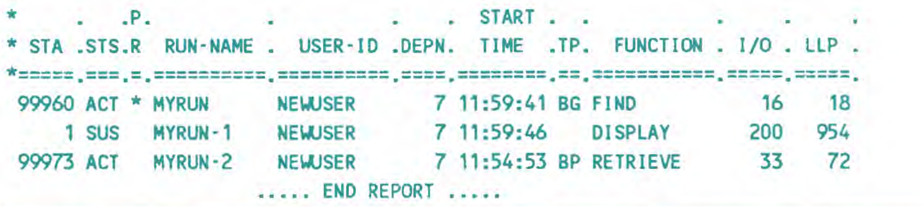
Procedure	Enter one of the following formats on the control line.	
Formats	**RS** Displays the function form.	
	RS[,*o*,*run*,*sn*] Displays status of runs.	
Format Fields	*o*	Options field. See Options.
	run	Run name (leave blank for the status of all runs).
	sn	Station number (for the status of only those runs at this station).

Options

Blank	All active and suspended runs
A	Active runs only
B	Background runs only
S	Suspended runs only

Example

Following is an example of RS information.



```
*     .   .P.          .          .    .  START .  .           .     .     .
* STA .STS.R  RUN-NAME .  USER-ID .DEPN.  TIME   .TP.  FUNCTION . I/O . LLP .
*=====.===.=.==========.==========.====.========.==.===========.=====.=====.
 99960 ACT * MYRUN      NEWUSER       7 11:59:41 BG FIND           16    18
     1 SUS   MYRUN-1    NEWUSER       7 11:59:46    DISPLAY        200   954
 99973 ACT   MYRUN-2    NEWUSER       7 11:54:53 BP RETRIEVE        33    72
                      ..... END REPORT .....
```

After the result displays, press the **Resume** function key to display a new result with updated status.

Explanation

Sta — Station number that started the run.

Sts — Run status. ACT = active, SUS = suspended, WAT = run is waiting.

Pr — Priority. Blank = normal, * = batch, A through G = Transaction Processing (TIP) priority level (A = 1, B = 2, ...G = 7).

Run-Name — Name of the run.

User-Id — User-id of user who started the run.

Depn — Department number of user who started the run.

Start Time — Time the run was started.

Tp — Type of run. Blank = normal, BP = batch port, BG = background, RR = remote (this field is not displayed if you specify the B option on the RS function).

Function — Last MAPPER function started at the station. If the run is suspended, the function may not have been started by the run.

I/O Current I/O count accumulated by the run.

LLP Current LLP count accumulated by the run. (LLP is the number of lines processed.)

The following fields are displayed when the B option is used:

Status Run status. ACTIVE = active, FACIL = background: waiting for background facilities to become available, COMPLT = scheduled background run is complete.

End Time Time the background run completes.

Org Sta Station number originating the background run.

Msg Sta Station number to notify when the background run completes.

Err Num MAPPER error number if the background run had an error.

S (Search)

Use the Search (S) function to find a character string in specified fields of one or more reports in a drawer and to create a result containing all lines on which the string is located.

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Was a report on display?
   Yes: Fill in the function mask.
   No: Fill in the function form then the function mask.

Menu Path

Select Task
    Find Data
        Search

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

S

Displays the function form.

S {*rd*[*c*]|*d*[*c*]} [*f*]

Displays the function mask.

Format Fields

rdc Report to search. (Or type **s -** to indicate the current report.)

1100: On OS 1100 MAPPER Systems, you do not need to type the minus (-) to indicate the current report.

dc Drawer to process. Use this field to search all reports in a given drawer.

f Report format in which to search for data (lets you look for data in fields beyond column 80 if you do not already have those columns on display).

Options

A — Processes all line types. Do not use this option with the T or U options.

B[(*n*)] — Stops a search after the *n*th find. Default = first find. The search continues until n items are found or until the end of the report.

C(S) — Distinguishes between uppercase and lowercase letters.

1100: C(x) — Alters the search process based on the character set order. Ordinarily the system processes the report based on the character set of the drawer. The C option allows you to choose the character set type on which to base the search. Use one of the following:

C(F)	Full character set (FCS)
C(L)	Limited character set (LCS)
C(S)	Strict comparison; distinguishes between uppercase and lowercase letters

D — Omits search information lines from the result. See the outcome section for the contents of search information lines.

1100: E*n* — Estimates the number of lines in the result, where *n* is an integer. This option improves the efficiency of the search when the output result contains more than 500 lines.

F — Searches for a numeric value instead of a character string. To search for a range of negative or floating point numbers, use this option.

H — In a multiple report search, includes the heading lines of the first report only.

L(x) — Omits lines of type *x* from the result. Note that you can use any type line here. For example, L(*.A) indicates that asterisk lines, period lines, and A-type lines are to be omitted from the result.

N — Includes only those lines not meeting the search criteria. Do not use this option with the T or U options.

continued

Options (cont.)

P — Extracts paragraphs from the report. A paragraph consists of data on a specified line type and all subsequent lines until the next occurrence of that line type.

For example, a normal search of asterisk lines creates a result listing asterisk lines as well as the trailer — period lines — that follow. With the P option, the result also includes tab lines that follow.

Q[(*n*)] — Extracts the first *n* paragraphs from the report; stops the search after the *n*th paragraph. Do not use this option with the A option. Default = first paragraph.

R*x*-*y* — Searches a range of reports from *x* through *y*.

R*x*,*y* — Searches reports *x* and *y*.

R*x*-*y*,*z* — Searches reports *x* through *y* and also *z*.

T[(*x*)] — Includes data found on the type of line you are processing and on the preceding *x* type line. Default = tab line. Do not use this option with the A, N, or U options.

A common use for the T option is to include the tab line preceding the asterisk line found.

U[(*x*)] — Includes data found in the type of line you are processing and its surrounding data unit. A data unit consists of a group of lines starting at line type *x* up to the next line of that type. Default = tab line. Do not use this option with the A, N, or T options.

With the U option, you specify the paragraph boundary as the *x* line type. In contrast, the P option automatically uses the line type you are processing as the paragraph boundary.

@ — Searches for spaces or a specific line type when used with the @ parameter.

/ — Searches for a slant (/) character as data when used with the / parameter.

Parameters

The character string you want to find is the most common parameter to use with the S function. You can also specify the following parameters:

/ Searches for a slant (/) character when used with the / option. Without the / option, a slant appearing in a parameter line shortens that field and a partial field is processed.

@ Searches for spaces. Use with the @ option: type @ in the columns of the function mask where you want to search for spaces. To find a blank field, type @ in the first column of the field. See the example "Searching for Spaces."

Also finds all lines of a specific type when used with the @ option. See the example "Searching for All Lines of a Certain Type."

R Searches for a range of data. See the example "Searching for a Range of Data."

Outcome

- At the top of the result the following information is displayed:
  - Number of finds
  - Number of lines searched
  - Search mask showing the options and parameters you specified

  Use the D option to omit this information from the result.

- A result is displayed containing the specified data including trailer lines, if any.

- If the specified data is not found, the search information lines state that no finds were made. If you used the D option, only a heading is displayed with no data beneath it.

Guidelines

Searching for More Than One Character String

To search for multiple strings in the same field, enter each character string on a separate line under the field heading in the mask.

To search for multiple strings in different fields, enter the character strings under different headings on the same line. Data lines are displayed in the result whenever all specified fields contain the requested strings.

To search for multiple strings in different fields and include either one string or the other or both, enter the character strings under different headings and on different lines in the mask. Data lines are displayed in the result whenever either of the specified fields contains the requested strings.

Searching for a Range of Strings

To search for a range of character strings, follow this procedure and refer to the accompanying example:

1. Type the characters for the lower end of the range in the first line under the mask in the correct field.
2. Type r in the first character position in the next line.
3. On the same line as the R parameter, type the characters for the higher end of the range in the same field as the lower end. Then transmit.

```
*St.Status.
*Cd. Date .
*==.======.
 ** ******
    800101   <-- Lower end of the range.
r   890520   <-- R parameter in column 1 designating a range
                 search; higher end of the range.
```

Searching on a Sorted Field

If the report is already sorted and you only need to search on the sorted field, you can use the Binary Find (BF) function with the O option instead of the S function. For large reports, this is more efficient than using the S function.

Hierarchy of Line Types

A Search function result lists line types according to this hierarchy:

1. Tab lines
2. Asterisk lines
3. Period lines

The result lists asterisk and period line types as trailer lines if they follow tab line types in the report. Similarly, the result lists period line types following asterisk line types.

Search Update Available

To create an update result that allows you to blend the changed lines back into the original report or delete lines from the original report, use the Search Update (SU) function. See the online help system (HELP,SU) for more information.

Searching for a Specific Line Type

To find all lines of a certain type, specify that line type as the type to process. For search parameters, specify a range search of every character in the ASCII character set. The first character to search for in the range is a space (use the @ option and parameter); the last character in the range is a tilde (~). See the example "Searching for All Lines of a Certain Type."

ISRH Run

1100: On OS 1100 MAPPER Systems, you can use the ISRH run to generate S function masks that you can save, alter, and use again. For information on how to use the iterative runs, see Section 5.

Examples

Example 1: Searching for Multiple Strings

To be included in the result of this example, a data line must contain both the characters ip and blackbox1 in the St Cd and Product Type fields or else both the characters sc and greenbox6 in those fields.

```
*St.Status.By. Product .
*Cd. Date .In.  Type   .
*==.======.==.=========.
 ** ****** ** *********
 ip           blackbox1
 sc           greenbox6
```

Example 2: Searching for a Range of Data

```
n
*St.Status.
*Cd. Date .
*==.======.
 ** ******
    830301
r   840603
```

n Display only the lines that do not meet the search criteria below.

830301
840603 Search for dates in this range in the Status Date field. Because the N option is specified, the result contains all lines containing dates outside this range.

r Search for a range of data.

Example 3: Searching a Range of Reports

```
hr2-5,19,100
*St.Status.
*Cd. Date .
*==.======.
 ** ******
 ip
```

`h` Include the heading lines of the first report only.

`r2-5,19,100` Search reports 2-5, 19, and 100.

`ip` Search for this character string in the St Cd field.

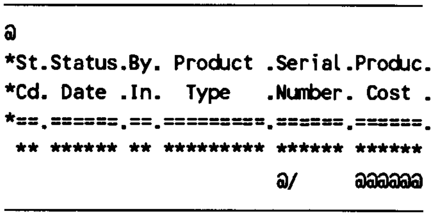
Example 4: Searching a Partial Field

```
*St.Status.By. Product .
*Cd. Date .In.  Type   .
*==.======.==.=========.
 ** ****** ** *********
              blackbox/
```

`blackbox` Search for this character string in the Product Type field.

`/` Stop processing the field at the column before this character. (You can achieve the same result by deleting the last asterisk in the Product Type field.)

Example 5: Searching for Spaces



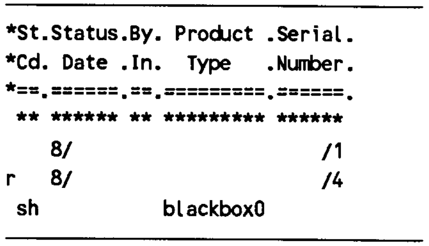
```
@
*St.Status.By. Product .Serial.Produc.
*Cd. Date .In.  Type   .Number. Cost .
*==.======.==.=========.======.======.
 ** ****** ** ********* ****** ******
                        @/     @@@@@@
```

@ Search for spaces.

@/
@@@@@@ Include in the result all lines containing both a space in the first column of the Serial Number field and a completely blank Product Cost field.

Example 6: Searching for Multiple Data in Range

To be included in the result of this example, a data line must contain either any status date in the 1980s and a serial number ending in 1, 2, 3, or 4, or else the characters sh and blackbox0 in the St Cd and Product Type fields.



```
*St.Status.By. Product .Serial.
*Cd. Date .In.  Type   .Number.
*==.======.==.=========.======.
 ** ****** ** ********* ******
    8/                      /1
r   8/                      /4
 sh           blackbox0
```

Example 7: Including the Previous Line Type

```
t
*St.Status.
*Cd. Date .
*==.======.
 ** ******
*sh
```

t Include the tab line preceding each of the found lines.

* Search asterisk lines.

sh Search for this string in the St Cd field.

Example 8: Omitting Certain Line Types

```
l(*A.)d
*St.Status.
*Cd. Date .
*==.======.
 ** ******
 ip
```

l(*A.) Omit asterisk, A, and period type lines from the result.

d Omit search information lines from the result.

ip Search for this string in the St Cd field.

Example 9: Searching for Specific Line Types

```
@
*St.Status.
*Cd. Date .
*==.======.
 ** ******
.@/
r~/
```

@ Search for spaces. You need this option for the range search described below.

. Search for period lines.

@/
r~/ Search for all characters in the ASCII character set: a range search (r) from space (@) to tilde (~). The slant (/) character shortens the field to process so that only the first character position of the St Cd field is searched.

Example 1Ø: Searching for Line Types in LCS Reports

1100: The following example applies only to OS 1100 MAPPER Systems.

As in the previous example, this example searches for a specific line type, but it uses an LCS report.

```
*St.Status.
*Cd. Date .
*==.======.
 ** ******
.@/
r./
```

. Search for period lines.

@/
r./ Search for all characters in the limited character set: a range search (r) from @ to period. The slant (/) character shortens the field to process so that only the first character position of the St Cd field is searched.

Example 11: Grouping Lines

1100: The following example applies only to OS 1100 MAPPER Systems.

```
sh
*St.Status.
*Cd. Date .
*==.======.
 ** ******
 sh
 sc
```

s Display all finds for each parameter separately.

h Display headings only once.

sh Display all lines containing sh first.

sc Display all lines containing sc second.

SEND (Send Report)

Use the Send Report (SEND) function to send an entire report from one station to another station. You can request that the system notify you when the recipient accepts the message. The recipient can also send a message back to you.

Prerequisite

The report you want to send must be on display.

How to Access the Function

Menu Procedure

Procedure	1. Select menu items using the menu path below. 2. Fill in the function form. For help with the function form, press **Help**.
Menu Path	Select Task Send and Receive Send a Report to a Station

Control Line Procedure

Procedure Enter one of the following formats on the control line.

Formats

SEND
Displays the function form.

SEND *sn*[*sl*,*ack?*]
Sends the report to the specified station.

Format Fields

sn Station number of the receiving station.

BTOS: Use the station name assigned by the MAPPER system coordinator.

sl Site letter of the receiving station. (For a list of sites, press **Remote** from the active screen.) Default = local site.

1100: This subfield is not available on OS 1100 MAPPER Systems.

BTOS: Multiple sites are not active concurrently.

ack? Request acknowledgement when the message is received? Y or N. Default = N.

Outcome

- The Send Report function sends the report as a result to the specified station. It redisplays the report on the sender's screen, along with a message indicating that the report was queued.
- The system creates a result when the receiver requests to display the message, in the cabinet and drawer of the original report being sent.
- Messages sent from a remote site reside as results in cabinet 0, drawer F. If the receiver of the message duplicates the result, the duplicate result resides in cabinet 0, drawer F.

1100: On OS 1100 MAPPER Systems, you cannot send messages to a remote site.

- The function displays a system message on your screen indicating that your message has been queued to the receiving station.

- The function notifies the receiving station that a message is waiting. See the Acknowledge Message (OK) function for more details.
- The system continues to attempt to deliver the message until the receiving user acknowledges the message.

Sign on to MAPPER Software

To access the MAPPER system, use the sign-on screen menu or the Sign On function format.

Prerequisite

Before you sign on, ask your supervisor or MAPPER system coordinator for a user-id and department number. Depending on how your site is configured, you may also need a password from your coordinator.

1100: On OS 1100 MAPPER Systems, if you see any words other than IDLE, PACK, or RECOVERY in the system logo, a session is already in progress. Someone has left the workstation without ending the MAPPER session.

Using NEWUSER

If you do not have your sign-on information from the coordinator, you can sign on as NEWUSER by pressing the **SignOn** key from the sign-on screen. This gives you access to the demonstration database for training purposes only.

How to Access the Function

Menu Procedure

Procedure

1. Fill in the fields on the sign-on screen menu, as listed in the Menu Fields below.
2. When you have finished filling in the fields, transmit.

Menu Fields

User-id A string of up to 11 characters, usually your initials or name. Your user-id must be registered by the MAPPER system coordinator.

Dept # Department number, which is a number designating a group of users. Your department number must also be registered by the coordinator.

Password A string of one to six alphanumeric characters used to prevent other users from using your sign-on. Depending on how your site is configured, you may need to obtain the password from your coordinator. You can change the password at any time.

Control Line Procedure

Procedure

With the sign-on screen on display, move the cursor to the home position and enter the following format. For security reasons, your sign-on does not appear on the screen as you enter it.

Format

]*user-id,dept*[,*psw*]

Format Fields

] A right bracket. This character must begin the sign-on.

user-id See menu field description (User-id).

dept See menu field description (Dept #).

psw See menu field description (Password).

Outcome

The MAPPER active screen is displayed. The system places you in the cabinet that is registered with your sign-on. If you wish to work in another cabinet, contact your coordinator and see the Cabinet Switch (C) function or the Cabinet Switch (CS) run.

Creating and Changing Passwords

To create or change a password, enter the following values in the Password field of the sign-on screen menu or function format. The password is one to six alphanumeric characters.

To do this...	Enter this in the Password field...
Create a new password	*/newpsw*
Change your password	*oldpsw/newpsw*
Clear your password	*oldpsw//*

Site Password Controls

This section on site password controls applies only to the OS 1100 MAPPER System.

Each site has the option of customizing their sign-on password controls. For more information call your coordinator about the Sign On function and the special password procedures in effect at your site.

SORT

Use the Sort function to put the lines of a report in a specified order. With the Sort function, you can do tasks including the following:

- Sort a report by placing a field in alphabetical order (ascending) or reverse alphabetical order (descending).
- Sort a report by organizing a field in numerical order. You can use the N parameter to sort numbers containing decimal points, spaces, plus, and minus signs.
- Sort a report using several fields. For example, you can sort the St Cd field in alphabetical order, and within that field, sort the By In field.

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Was a report on display?

   Yes: Fill in the function mask.

   No: Fill in the function form then the function mask.

Menu Path

Select Task
    Move Data
        Sort a Report

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

SORT

Was a report on display?

Yes: Displays the function mask.

No: Displays the function form.

SORT *rd*[*c f*]

Displays the function mask.

continued

Control Line Procedure (cont.)

Format Fields (cont.)	*rdc*	Report to sort.
	f	Report format in which to display the function mask.

Options

A — Processes all line types.

C(S) — Distinguishes between uppercase and lowercase letters. Uppercase letters precede lowercase letters on an ascending sort. For example, Zebra precedes apple.

1100: C(*x*) — Alters the sort process based on the character set order. Ordinarily the system processes the report based on the character set of the drawer. The C option allows you to choose the character set type on which to base the sort. Use one of the following:

C(F)	Full character set (FCS)
C(L)	Limited character set (LCS)
C(S)	Strict comparison; distinguishes between uppercase and lowercase letters

1100: X+ — Sorts the data using the OS 1100 Sort/Merge product (if installed at your site), regardless of the report length.

1100: X- — Sorts the data using the normal MAPPER software process, regardless of the report length.

If you do not specify X+ or X-, the sort process is determined by the report length. See the coordinator for more information.

Parameters

Note that you can sort fields totaling 68 characters in width.

1 - 5	Specifies the fields to sort and the order in which to sort them. 1 is the highest level and 5 is the lowest level.
D	Sorts in descending order. Type the D following the level number 1 through 5, for example, **2d**.
N	Sorts fields containing numbers with decimal points, spaces, plus and minus signs. Type the N following the level number, for example, **1n**.

Outcome

- **1100:** The system places an update lock on the report to prevent other users from updating it while it is sorting.
- A result is displayed with fields sorted as you specified.
  Default = ascending order.
- The function sorts the text according to the alphabetic order of the language used.

1100: The function sorts the text according to the character set order of the drawer (or the character set order you specify if you use the C(*x*) option). For more information, see Appendix B.

Guideline

ISOR Run

1100: On OS 1100 MAPPER Systems, you can use the ISOR run to generate SORT function masks that you can save, alter, and use again. For information on how to use the iterative runs, see Section 5.

Examples

Example 1: Sorting in Ascending and Descending Order

This example sorts the St Cd field in ascending order and the Product Type field in descending order. Here is the function mask used in this example:

```
a
*St.Status.By. Product.
*Cd. Date .In.  Type  .
*==.======.==.========.
 ** ****** ** ********
 1             2d
```

a Specifies that all line types are to be processed.

1 Identifies St Cd as the primary sort field (to be sorted in ascending order).

2d Identifies Product Type as the secondary sort field (to be sorted in descending order).

Example 2: Sorting Numbers

This example uses the N parameter to sort numbers that contain decimals and negative signs. Here is a portion of the sample data for this example. Note that the numbers in this sample field vary in form and position.

```
*====.
 123
  1.2
 -12
   4
```

This is a portion of the function mask used in this example:

```
*====.
 ****
 1n
```

This is a portion of the result for the example:

```
*====.
 -12
  1.2
    4
 123
```

SS (Station-to Station Message)

Use the Station-to-Station Message (SS) function to send data or a message to a configured MAPPER station. Send a maximum of one screen of data.

Prerequisite

To send part of a report, that part of the report must be on display before you request the function. (Note that you can send part of a report only by entering the SS function format on the control line; requesting the function through the menus clears the screen.)

Note: Do not send system messages or data containing SOE characters.

How to Access the Function

Menu Procedure

Procedure	1. Select menu items using the menu path below. 2. Follow the procedure in "Creating and Sending the Information" in this section.
Menu Path	Select Task Send and Receive Send a Message to a Station

Control Line Procedure

Procedure	Enter one of the following formats on the control line.
Formats	**SS** Retains the current screen and requests information at the top of the screen. You can send messages and data, including lines from the displayed report. **SS,C** Clears the screen and requests information at the top of the screen. You can type a message on the blank screen.
Format Field	**C** Clear the screen.

Outcome

- The function places the message into a result in cabinet 0, drawer F. (If the receiver duplicates the result, the duplicate result resides in cabinet 0, drawer F.)

1100: On OS 1100 MAPPER Systems, the function places the message into a result in cabinet 0, drawer A.

- The SS function displays a system message on your screen indicating that your message or data has been queued to the receiving station.
- The function notifies the receiving station that a message is waiting. See the Acknowledge Message (OK) function for more details.

  The system continues to attempt to deliver the message until the receiving user acknowledges the message.
- If you typed Y in the **Request message acknowledge** field, the system sends a message to you indicating that the receiving user has acknowledged the message.

Creating and Sending the Information

When you request the SS function, two fields are displayed on the top line of your screen. Follow these steps:

1. In the **Message to station** field, type the station number and the site letter (optional) of the receiving station. (To see a list of sites, press **Remote** from the active screen.)

▽ **1100:** On OS 1100 MAPPER Systems, you cannot send messages to a remote site.

▽ **BTOS:** Multiple sites are not active concurrently.

2. To request an acknowledgment when the message is received, tab to the **Request message acknowledge** field and type **y**.

3. To send data, move your cursor below the top line and type your message, up to one screen.

4. Transmit.

Note: *If you entered SS on the control line with a report on display, you can also send part or all of the displayed report. Move the cursor below the part to be sent and transmit.*

START

This section does not apply to the BTOS II MAPPER System.

Use the Start function to collect data in a MAPPER report from one or more MAPPER reports and use it as input to an external task or to process as a script. You can also use it to start jobs that produce files you can retrieve into the MAPPER system with the Retrieve File (RET) function.

U Series: For a batch run, the Start function executes UNIX shell or IBM remote job entry (RJE) commands contained in a MAPPER report.

1100: The Start function executes a MAPPER runstream with Executive control language (ECL) statements as a batch run.

PC MAPPER: The Start function executes the command shell (CMD.EXE) contained in a MAPPER report.

Prerequisite

Data to be passed to a job must exist in a MAPPER report.

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Fill in the function form. For help while filling in the function form, press the **Help** key.

Menu Path

```
Select Task
    Use Utilities
        Start a Run
            Start a Batch Run
```

Control Line Procedure

Procedure

Enter the following format on the control line.

Format

START
Displays the function form.

Guidelines

Passing Reports and Drawers

To pass more than one report or drawer, place the $INCL$ command in the report containing the runstream you are starting. See Appendix D.

Returning Files to the MAPPER System

Use the Retrieve File (RET) function to retrieve files produced by the job, or use the MAPPER sign-on command **mapper -f** to pass a file containing a BPRUN$ command to the MAPPER system for execution.

See other parts of this section for BPRUN$ and **mapper -f** formats.

1100: Use the Retrieve File (RET [P]) function to retrieve files produced by the job, or use SYM statements and BPRUN$ commands to bring in the data one report at a time.

A Series: Use the Retrieve (RET) function to retrieve files produced by the job, or use the following command to execute MAPPER software as a program and pass a file containing a BPRUN$ statement for MAPPER software to execute:

```
RUN *SYSTEM/MAPPER;FILE PARAMS(NOTE="-Ssl -Ffilename");
```

where:

-S*sl* Designates the site; *sl* is the site letter.

-F*filename* Designates the file; *filename* is the title of the file containing a BPRUN$ command to send to MAPPER software (case sensitive).

If an interface name is used, a file title containing the data is given as a STRING parameter to the job.

BPRUN$ Command

The BPRUN$ command starts a run on the MAPPER system and optionally returns data from a job to the MAPPER system.

Include the BPRUN$ command in the file being sent to the MAPPER system. Build the BPRUN$ command into the output of a job, and start the command in column 1.

Within the output file, use the following format for the BPRUN$ command:

```
BPRUN$[,sl,cont] run[,data] user,dept[,pw c,cpw,d,f,ltyp,lsp?
ΔBPERR,edata]
```

1100:

```
BPRUN$[,cont] run[,data] user,dept,pw [c,cpw,d,f,ltyp,lsp?
ΔBPERR,edata]
```

where:

BPRUN$ Batch run command.

sl Site letter. Default = local site.

cont A continuation character to indicate that the statement is continued on the next line. Use any character that does not appear elsewhere in the BPRUN$ command. Also type this character at the end of the first line of the BPRUN$ command data

run Name of the run to start in the MAPPER system.

Note: *The run cannot contain any DSG, DSM, DSP, DSX, GOC, GS (when display is requested), HST, ITV, OTV, OUM, OUT, OUV, RDB, REL, SC, XIT, or XUN statements.*

1100:

Notes:

1. *The run cannot contain any DSG, DSM, DSP, DSX, GOC (when display requested), GS (when display requested), ITV, OUM, OUT, OUV, REL, RSI, SC, or XIT statements.*
2. *You can use the DSG, OUT, and SC statements if you send the output to another terminal.*

data Data passed to the MAPPER run and picked up by INPUT$ in the run. Separate multiple data items with commas and do not include spaces within or between data items.

user User sign-on.

dept User department number.

pw User sign-on password, if applicable.

c Cabinet in which to place the passed data.

cpw Cabinet password.

d Drawer letter.

f Format.

ltyp Line type for tab character insertion. Type Y to replace spaces with tab characters in those column positions defined in report 0 having tab characters; type a predefined line type number instead of Y for a predefined line defined in report 0.

lsp? Retain blank lines in the file? Y or N. Default = N, delete blank lines.

BPERR Name of the run to execute if an error occurs during the batch processing. (Call your coordinator to see if your site has a modified version of the BPERR run to handle batch errors.)

edata Data passed to the BPERR run and picked up by INPUT$ in the BPERR run. Type the station number to which to direct BPERR run error messages as the first data item. Separate multiple data items with commas and do not include spaces within or between data items.

The specified run starts in the MAPPER system.

If you specify a cabinet, cabinet password, and drawer, the data returned may be referenced as -0.

If the BPRUN$ run cannot sign on (invalid user-id or department), the system sends the BPERR message to the coordinator. If the BPRUN$ command contains and error but the sign-on is correct, the system executes the BPERR run. If an error occurs in the run specified in the BPRUN$ command, the system sends a message to the user-id on the BPRUN$ command.

Using the BPRUN$ Command

This section on using the BPRUN$ command does not apply to the OS 1100 MAPPER System. See "Using the BPRUN$ Command: OS 1100" in this section for a description of the BPRUN$ command as it applies to the OS 1100 MAPPER System.

To send native data to the MAPPER system, include the following command:

```
mapper [-ssl] -ffilename
```

where:

mapper Executes MAPPER software (must be lowercase).

-s MAPPER software option that indicates that a site letter follows.

sl Site letter. Default = local site.

-f MAPPER software option that indicates that the name of a file to process follows.

filename Complete path name of the file containing a BPRUN$ command to send to the MAPPER system (case sensitive).

If the host detects a BPRUN$ command in the file, it sends the file to the MAPPER system for processing.

U Series: All IBM output files automatically go to the MAPPER system so you do not need to include a **mapper -f** statement in the job stream.

U Series: See the example of a UNIX shell script returning output to the MAPPER system later in this section.

A Series: To send output from a job to MAPPER software, the following command must be included in this format:

```
RUN *SYSTEM/MAPPER;FILE PARAMS(NOTE="-Ssl -Ffilename");
```

where:

-S*sl*	Designates the site; *sl* is the site letter.
-F*filename*	Designates the file; *filename* is the title of the file containing a BPRUN$ command to send to MAPPER software (case sensitive).

If the BPRUN$ command is found, the file is sent to MAPPER software for processing.

Using the BPRUN$ Command: OS 1100

This section on using the BPRUN$ command applies only to the OS 1100 MAPPER System. See "Using the BPRUN$ Command" in this section for a description of the BPRUN$ command as it applies to other MAPPER systems.

Queue the file to the MAPPER interface with the following Executive control language (ECL) statement:

```
@SYM fn,,dev
```

where:

fn	Name of the file.
dev	Any valid symbiont device identification for the MAPPER system (normally MAPPER).

Requirement

The symbiont device must be configured as a batch port name to the MAPPER system, and as a local device to the OS 1100 operating system.

If the BPRUN$ command is found, the file is sent to MAPPER software for processing.

ENDRD$ Command

The ENDRD$ command (or another BPRUN$ command) marks the end of data to include in the report and indicates to the MAPPER system that the specified run is to start.

Enter the ENDRD$ command starting in column 1 of the file.

If the host does not detect an ENDRD$ command, it sends the entire file (after the BPRUN$ command).

When using several BPRUN$ commands in a file, you do not need to include ENDRD$ commands to mark the end of the preceding segments of data since subsequent BPRUN$ commands mark the end of the data.

The host processes all BPRUN$ commands imbedded in a file.

IBM Start Requirements

This section on IBM Start Requirements applies only to the U Series MAPPER System and the UNIX MAPPER Systems.

The IBM binary synchronous communications (BSC) and system network architecture (SNA) interface capabilities incorporated into the MAPPER system let you start IBM job streams on an IBM mainframe through BSC and SNA protocol. The bscsym or snasym shell scripts return files from the IBM mainframe to the MAPPER system.

If an error occurs in the job stream and the BPRUN$ through ENDRD$ does not get built, the bscsym or snasym shell script places a BPRUN$ command into the output file and starts the BPERRIBM run to send the output to the MAPCOORD user-id at the default site.

BSC and SNA daemon programs transfer the job streams and host files. How often the daemon runs depends on the entry in the crontabs directory put in by the UNIX system administrator during installation. When the daemon finishes transferring job streams and receiving output files from the IBM host, the host starts any runs specified by the BPRUN$ command.

If you want output from the execution of the IBM job stream, place an IBM ROUTE statement in the job stream. See the UNIX system administrator for the remote address of the BSC and SNA connection.

See the example of an IBM SNA job stream returning output to MAPPER software.

Examples

These examples are only for the U Series MAPPER System and the UNIX MAPPER Systems.

Example 1: UNIX Shell Script Returning Output

This is an example of a UNIX shell script to return output to MAPPER software:

```
cat > /usr/work/newuser/filea<<end*of*input
bprun$,a getfile,18 newuser,7,newuser 0,open,a BPERR,18
======== Line 1 of output file ===========
         Line 2 of output file
         Line 3 of output file
         Line 4 of output file
         Line 5 of output file
endrd$
end*of*input
mapper -sa -f/usr/work/newuser/filea
```

The data following the **cat** command is redirected to the file **/usr/work/newuser/filea**. All data between the END*OF*INPUT statements is included in the file (including the BPRUN$ and ENDRD$). The file is then sent to MAPPER site A via the **mapper -f** statement.

When this file is sent to MAPPER software, the GETFILE run is executed.

Example 2: IBM SNA Job Stream

This is an example of an IBM SNA job stream to return output to MAPPER software:

```
/*ROUTE     PRINT RMT19
//*         CREATE BPRUN$ FOR 'PRINTED' OUTPUT
//          EXEC  PGM-IEBGENER
//SYSPRINT  DD    DUMMY
//SYSIN     DD    DUMMY
//SYSUT2    DD    SYSOUT=A,DCR=BLKSIZE=133
//SYSUT1    DD    *
BPRUN$,A    GETFILE,18 NEWUSER,7,NEWUSER 0,OPEN,B BPERR,18
/*
//TESTPRT   EXEC  PGM=IEBGENER
//SYSPRINT  DD    DUMMY
//SYSIN     DD    DUMMY
//SYSUT2    DD    SYSOUT=A,DCB=BLKSIZE=133
//SYSUT1    DD    *
===========FIRST LINE OF FILE ============
LINE 1 OF EXAMPLE
LINE 2 OF EXAMPLE
LINE 3 OF EXAMPLE
LINE 4 OF EXAMPLE
LINE 5 OF EXAMPLE
LAST LINE OF EXAMPLE
/*
//*         CREATE ENDRD$ FOR END OF DATA
//          EXEC  PGM=IEBGENER
//SYSPRINT  DD    DUMMY
//SYSIN     DD    DUMMY
//SYSUT2    DD    SYSOUT=A,DCB=BLKSIZE=133
//SYSUT1    DD    *
ENDRD$
/*
```

The output from the job stream (including the BPRUN$ and ENDRD$ commands) is sent to MAPPER software. The GETFILE run is executed on MAPPER site A.

STOP

This section applies only to the OS 1100 MAPPER System.

Use the Stop function to stop any or all background runs being executed under your user-id.

How to Access the Function

Control Line Procedure

Procedure	Enter one of the following formats on the control line.	
Formats	**STOP** Displays the function form.	
	STOP,*run*[,*sn*] Terminates the run and returns a message confirming the termination.	
Format Fields	*run*	Name of the background run you wish to abort.
	sn	Station number that started the run.

Outcome

The system terminates the background run you specify.

SV (Save Report Version)

Use the SV (Save Report Version) run to temporarily save a report or reult. Here are the tasks you can do with this function:

- Retain a result while displaying another report.
- Copy a report to a different cabinet. For example, you may want to switch cabinets, examine the contents of the new cabinet to decide where to place the report, then copy the saved report.
- **1100:** On the OS 1100 MAPPER System the SV run can be used to save up to four reports.

Prerequisite

A report must be on display.

How to Access the Run

Menu Access Procedure

Procedure	Select menu items using the menu path below.
Menu Path	Select Task Use Utilities Store Report Temporarily **1100:** Save Results Temporarily

Control Line Procedure

Procedure	Enter the following format on the control line.
Format	**SV** Redisplays the report and saves it as a temporary result.

Copying a Report to a Different Cabinet

You can use the Replace Report (REP) function to copy a report to a different cabinet; see the REP function for more information.

However, if you do not know which report or drawer in which to place the report, use the SV run. It allows you to save the report as a result, switch to the new cabinet, and examine its contents before saving the report. This procedure is described here:

1. With the report to copy on display, use either the menu procedure or the control line procedure to execute the SV run.
2. Switch to a different cabinet using the Cabinet Switch (C) function. (Do not use the CS run.)
3. If necessary, examine the Drawer Select menu to decide where to place the report.
4. Press **Resume** to display the saved result.
5. Save the result as a report using the Replace Report (REP) or Duplicate Report (XR) functions.

Caution

Between steps 1 and 4, do not use any function that can be resumed or use any run. If you do, the system may release the saved version.

Saving Four Reports or Results

This section on saving four reports applies only to the OS 1100 MAPPER System.

Use the SV run to save up to four reports or results using the function keys **F1** through **F4**.

Outcome

The function key bar is redefined. The number of the saved report or result is displayed with the corresponding function key (1 through 4) or the word Open is displayed if nothing is saved. Pressing the **Contrl** function key redisplays the original function key definitions. To regain the SV function key bar, press **Resume**.

Procedure

1. Use the SV run to save the first report or result.
2. Display the second report or result and press **F2**.
3. Display the third report or result and press **F3**.
4. Display the fourth report or result and press **F4**.

To redisplay these saved results, press the related function key.

SYSTEM

This section does not apply to the BTOS II MAPPER System or the OS 1100 MAPPER System.

Use the System function to display a result listing the current system activity.

How to Access the Function

Menu Procedure

Procedure	Select menu items using the menu path below.
Menu Path	Select Task Use Utilities Display a List of Current Users

Control Line Procedure

Procedure	Enter the following format on the control line.
Format	**SYSTEM** A result is displayed listing the current system activity.

Outcome

The function displays a result containing the following information:

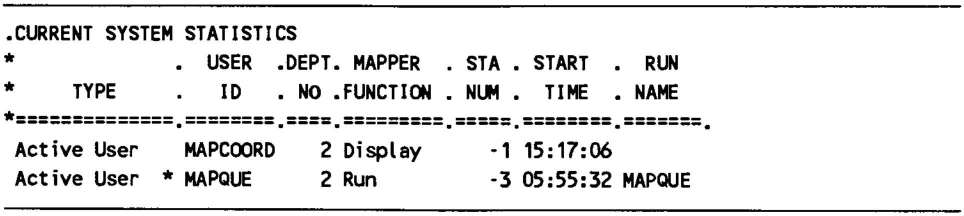
- Active users
- Active runs and functions
- Active stations
- The time the last run or function started
- The run (if any) being executed at each station

Displaying Again

To display a new result reflecting the most current activity, press **Resume**.

Example

This is a sample result showing the current system activity:



```
.CURRENT SYSTEM STATISTICS
*              .  USER  .DEPT. MAPPER  . STA . START  .  RUN
*     TYPE     .   ID   . NO .FUNCTION . NUM .  TIME  . NAME
*==============.========.====.=========.=====.========.=======.
 Active User    MAPCOORD    2 Display      -1 15:17:06
 Active User  * MAPQUE      2 Run          -3 05:55:32 MAPQUE
```

Note that an asterisk (*) in the last column of the Type field indicates that the station is busy.

TOT (Totalize)

Use the Totalize (TOT) function to perform arithmetic and move operations on fields in reports. These are a few operations the TOT function performs:

- Adding, subtracting, multiplying, or dividing values by a constant value
- Computing horizontally and vertically
- Subtotaling fields when key fields change
- Changing the line type
- Counting entries
- Copying data from one field to another field or fields

How to Access the Function

Menu Procedure

Procedure

1. Select menu items using the menu path below.
2. Was a report on display?
   Yes: Fill in the function mask.
   No: Fill in the function form then the function mask.

Menu Path

Select Task
    Perform Math Operations
        Totalize Data

Control Line Procedure

Procedure

Enter one of the following formats on the control line.

Formats

TOT
Was a report on display?
Yes: Displays the function mask.
No: Displays the function form.

TOT *rd*[*c f*]
Displays the function mask.

continued

Control Line Procedure (cont.)

Format Fields

rdc Report to process.

f Report format in which to totalize data (lets you process data in fields beyond column 80 if you do not already have those columns on display).

Options

A Processes all line types.

C Displays a subtotal if the value in the subtotal key field changes case. For example, with the C option, if the value in the key field is Rate on the third line and RATe on the fourth, a subtotal is shown after the third line. Use this option with the S parameter only.

E Counts entries (see also the R option).

H Cumulates horizontally.

I Ignores headings: processes the report even though asterisk lines or the heading divider line may not exist.

J(*x*) Justifies numeric result values in fields to *x*; *x* is one of the values listed below. When displayed, leading zeros appear to the left of a value, nonsignificant zeros to the right. Neither affect the value.

- c Eliminates leading and nonsignificant zeros. Inserts commas every three digits in the integer portion of the result. Right-justifies values.
- l Eliminates leading and nonsignificant zeros. Left-justifies values.
- r Eliminates leading and nonsignificant zeros. Right-justifies values.
- x Eliminates leading zeros. Left-justifies values, and fills fields with nonsignificant zeros.
- z Eliminates leading and nonsignificant zeros. Right-justifies values, and fills leading spaces with zeros.

continued

Options (cont.)

N Displays three blank lines between subtotal groups and omits the grand total. Use with the S option.

O Omits all data lines from the result, displaying only subtotal and grand total information.

R*n* Rounds numbers to the nearest *n*. Use with the E option to round entry counts. The range for *n* is .0000000000000001 to 100000.

For example if you type r.001, the values are rounded to the nearest thousandth. If you type r10000, the values are rounded to the nearest 10 thousand.

S Places subtotals in vertical operation fields.

=*x* Changes the line type indicator in column 1 to *x*, where *x* is an asterisk, period, or alphanumeric character.

* Omits asterisk flags following results of vertical operations, except for grand totals, when some of the data was invalid (nonnumeric or blank).

Parameters

You can perform up to 16 operations in any one parameter line, and you can use up to two parameter lines.

+ Addition, flagging the field to operate on (for example, flagging the field to divide into by placing + in that field)

- Subtraction

* Multiplication

/ Division

= Totalling horizontally, filling fields, numbering sequentially

A Averaging (Blanks are not considered zeros.)

C Cumulating data or sequencing data lines

continued

Parameters (cont.)

D	Deleting commas
E	Counting entries and displaying the number at the end of result
I	Inserting commas
M	Copying data
S	Subtotalling (specifies subtotal key fields)
S1 - S5	Hierarchical subtotalling

Outcome

- The function processes the operations and creates a result.
- Results of vertical operations appear at the end of the result.
- Whenever at least one field item processed in a vertical operation contains nonnumeric data or no value (blank), an asterisk appears next to the resulting value as a flag indicating that some of the data was invalid.

Guideline

ITOT Run

1100: On OS 1100 MAPPER Systems, you can use the ITOT run to generate TOT function masks that you can save, alter, and use again. For information on how to use the iterative runs, see Section 5.

Computing Horizontally Using the Totalize Function

Use arithmetic operators (+, -, *, /) to do computations in fields on each line. Use the = parameter to store the resulting values in a field on the same line.

Note: *You can perform only one horizontal operation at a time.*

To compute horizontally, follow this procedure and see the example "Computing Horizontally."

1. Type + in the field containing the data to add to, subtract from, multiply, or divide into.
2. Type + in the field to add, - in the field to subtract, * in the field to multiply by, or / in the field to divide by.
3. Type = in the field you want to contain the result.

To receive a grand total at the end of the report, type + in the line directly below the equal sign.

Computing Horizontally with Constant Values Using Totalize

To use a constant while computing horizontally, follow this procedure and see the example "Computing Horizontally Using Constant Values."

1. Type the operator (+, -, *, /) and the constant value in the field you want to process. To subtract a constant value, type a plus sign before the minus sign. For example, type **+-2** to subtract 2 from a field.
2. Type = in the field you want to contain the result. To place the result in the same field, type = in the line below the operator in that field.

Note: *If the constant value includes a decimal, precede the decimal point with a zero; for example, 0.1.*

Copying Fields Using Totalize

Use the M and = parameters to copy data to another field or fields. Follow this procedure and see the example "Copying Fields Using Totalize":

1. Type m in the issuing field, the field to copy.
2. Type = under the first asterisk in the receiving field, the field you want to contain the data.

The receiving fields must be the same width as the field to be copied. Add or erase asterisks under the field headings to make the fields the same size.

Filling Fields with Data Using Totalize

Use the = parameter to fill fields with specified data. The = parameter offsets the specified data by one character, so the function considers the following tab in the mask as part of what you are inserting unless you erase it.

Use the following procedure, and see the example "Filling Fields Using Totalize."

1. Type = under the asterisk of the first column in a field to fill.
2. Type the data to be filled in the field.
3. Erase the tab character in the mask that follows the field you are filling.

Note: *Other fields cannot have arithmetic or move symbols.*

To fill a field with spaces, type = under the first asterisk in the field and erase the next tab character in the mask.

Subtotaling Data Using Totalize

Use the S option and the S and + parameters to display subtotals vertically at each line where a specified field, the subtotal key field, changes values. Follow this procedure and see the example "Subtotaling Data Using Totalize":

1. Type **s** in the options field above the mask.
2. Type **s** in the subtotal key field.
3. Type **+** in one or more fields you want added. Each time the value in the key field changes, the addition starts over.

You can also average fields by using a subtotal key. Simply use the A parameter instead of the + parameter. The function averages the data in each group of subtotal keys.

The result includes three lines inserted between each subtotal key group:

- An asterisk line with dashes in the fields you added or averaged
- Another asterisk line containing the word SUBTOTAL below each subtotal key (or part of the word SUBTOTAL if the key field is fewer than eight characters) and the subtotals or averages under the fields processed
- A blank period line to separate each subtotal group

To display only the subtotals and grand total, use the O option with the S option. The last line from each subtotal key group appears in the result along with the subtotals for the fields.

Combining Subtotal Fields Using Totalize

You can combine up to five subtotal key fields so that whenever the values in any of the fields change, the subtotaling starts over. Follow this procedure and see the example "Combining Two Subtotal Fields" in this section.

1. Type `s` in the options field above the mask.
2. Type `s` under the subtotal key fields you want to combine.
3. Type `+` in one or more fields you want added. Each time the values in any of the key fields change, the addition starts over.

Hierarchical Subtotaling Using Totalize

You can assign a hierarchy of up to five key fields (S1 through S5). Whenever the value changes in a lower-level key field such as S2, the subtotal is displayed. Whenever the value changes in a higher-level key field such as S1, the subtotal of the previous subtotals is displayed.

If a higher-level key field stays the same but a lower-level key field changes, even though a subtotal is displayed, the subtotaling for the higher-level continues as long as its value stays the same.

Use the same procedure as other subtotaling operations, except type `s1` through `s5` under the key fields you want to subtotal hierarchically. See the example "Hierarchical Subtotaling."

Examples

Example 1: Computing Horizontally

This is the function mask used in this example:

```
* Product .         .Space.  Demo  .               .
*  Type   .         . Req .Quantity. Demo Results  .
*=========.  . . .  .=====.========.===============.
 *********           ***** ******** ***************
                       +     -        =
```

+ Subtracts from this field.

- Identifies the field to subtract.

= Places the result in this field.

Example 2: Computing Horizontally Using Constant Values

This is the function mask used in this example:

```
* Product .         .Space.  Demo  .               .
*  Type   .         . Req .Quantity. Demo Results  .
*=========.  . . .  .=====.========.===============.
 *********           ***** ******** ***************
                           /2
                           =
```

/2 Divides all of the values in this field by 2.

= Places the result in this field (the same field as the one being divided).

Example 3: Copying Fields Using Totalize

This is the function mask used in this example:

```
* Product .         .Space.  Demo  .               .
*  Type   .         . Req .Quantity. Demo Results  .
*=========.  . . .  .=====.========.===============.
 *********           ***** ** ***** ********* *****

                     m        =               =
```

m Copies this field.

= Identifies the fields in which to copy the data.

Note: *The receiving fields are the same size as the field being copied. Asterisks were erased to make the fields the same size.*

Example 4: Filling Fields with Data

This example uses a vertical bar to indicate tabs not erased from the function mask and tabs that appear in the result.

This is the function mask used in this example. Note that the tab following the string abcd was not erased.

```
* Product .         .Space.  Demo  .               .
*  Type   .         . Req .Quantity. Demo Results  .
*=========.  . . .  .=====.========.===============.
 *********           ***** ******** ************* *
|=                  |=  5  =abcd   |              =1
```

Here is the result of this example. Note that since the tab following the abcd string was not erased, an extra tab appears in the result:

```
* Product .          .Space.  Demo   .              .
*  Type   .          . Req .Quantity. Demo Results  .
*=========.  . . .   .=====.========.==============.
|         |          |  5  |abcd    ||            1
|         |          |  5  |abcd    ||            1
|         |          |  5  |abcd    ||            1
```

Example 5: Subtotaling Data Using Totalize

This is the function mask for this example:

```
s
* Product . Sub .          .   Demo    .
*  Type   . Key .          . Quantity .
*=========.=====.  . . .  .==========.
 ********* *****           **********
           s               +
```

s Option displays a subtotal whenever the subtotal key field changes.

s Parameter identifies the subtotal key field.

+ Adds the values in this field until the key field changes, then starts the addition over.

This is the result of this example:

```
* Product . Sub .         .   Demo   .
*  Type   . Key .         . Quantity .
*=========.=====.  . . .  .==========.
 BLACKBOX3     A                   7
 BLACKBOX5     B                  60
 BLACKBOX7     C                 110
```

Example 6: Combining Horizontal Math and Subtotaling

This example divides the values in the Space Req field by those in the Demo Quantity field and keeps a running total of the division results in the Demo Results field. When the value in the Sub Key field changes, the running total starts over.

This is the function mask for this example:

```
so
* Product . Sub .       .Space.   Demo    .               .
*  Type   . Key .       . Req . Quantity . Demo Results   .
*=========.=====. . . . .=====.==========.===============.
 ********* *****         ***** ********** ***************
           s             +     /          =
```

s Option requests the subtotal operation.

o Displays the last line of each subtotal key group as well as the subtotal.

s Parameter identifies the key field.

+ Identifies the field to divide into.

/ Identifies the field to divide by.

= Identifies the field to place cumulated division results.

This is the result of this example:

```
* Product . Sub .        .Space.  Demo     .                 .
*  Type   . Key .        . Req . Quantity . Demo Results    .
*=========.=====. . . . .=====.=========.===============.
 BLACKBOX1     A          100          1            100
 BLACKBOX2     A          110          2            155
 BLACKBOX3     A          120          4            185
 BLACKBOX4     B          130         10             13
 BLACKBOX5     B          140         50 15.800000000000
```

Example 7: Combining Two Subtotal Fields

This example combines the fields Codel and Code2. When the value in either of them changes, a subtotal is displayed and subtotaling begins fresh.

Sample database:

```
*Code1.Code2.Number.
*=====.=====.======.
 one   a         22
 one   a          2
 two   a          5
 two   b          5
 two   b          4
```

Function mask:

```
s
*Code1.Code2.Number.
*=====.=====.======.
 s     s     +
```

Result:

```
*Code1.Code2.Number.
*=====.=====.======.
 one   a         22
 one   a          2
*----- ----- ------
*one   a         24
.
 two   a          5
*----- ----- ------
*two   a          5
.
 two   b          5
 two   b          4
*----- ----- ------
*two   b          9
.
*GRAND  -        38
```

Example 8: Hierarchical Subtotaling

This example displays a subtotal of the Number field when the Code2 field changes. It also displays a subtotal of the previous Number subtotals when the Code1 field changes.

Sample database:

```
*Code1.Code2.Number.
*=====.=====.======.
 one   a         22
 one   a          2
 one   a          2
 two   a          5
 two   a          5
 two   b          5
 two   b          4
```

Function mask:

```
s
*Code1.Code2.Number.
*=====.=====.======.
 s1    s2    +
```

Result:

```
*Code1.Code2.Number.
*=====.=====.======.
 one   a         22
 one   a          2
 one   a          2
*----- ----- ------
*      a         26
*one             26
.
 two   a          5
 two   a          5
*----- ----- ------
*      a         10
.
 two   b          5
 two   b          4
*----- ----- ------
*      b          9
*two             19
.
*GRAND  -        45
```

Example 9: Rounding Result Values

The function rounds the totals of the Space Req and Demo Quantity fields to the nearest thousandth and displays the them at the end of the result.

```
r.001
* Product .         .Space. Demo    .                 .
*  Type   .         . Req .Quantity. Demo Results    .
*=========.  . . .  .=====.========.==============.
 *********           ***** ******** ************* *
                      +     +
```

r.001 Rounds result values to the nearest thousandth.

+ Vertically adds the values in these fields.

Example 1Ø: Numbering Lines

```
e
* Product . Sub .
*  Type   . Key .
*=========.=====.
 ********* *****
          c
```

e Counts data lines of the type you are processing.

c Numbers the data lines sequentially in this field starting with 1.

This is the result of this example:

```
* Product . Sub .
*  Type   . Key .
*=========.=====.
 BLACKBOX1     1
 BLACKBOX2     2
 BLACKBOX3     3
 BLACKBOX4     4
```

UNIX (UNIX Interface)

This section applies only to the U Series MAPPER System and the UNIX MAPPER Systems.

Use the UNIX (UNIX Interface) function to access the UNIX operating system and execute UNIX commands.

Prerequisite

This function should be used only by experienced UNIX users.

How to Access the Function

Menu Procedure

Procedure	Select menu items using the menu path below.
Menu Path	Select Task Use Utilities Access Operating System

Control Line Procedure

Procedure	Enter the following format on the control line.	
Format	**UNIX [*o*] [*cmd* [*args*]]**	
Format Fields	*o*	Options field. See Options.
	cmd	UNIX command to be executed; a standard UNIX search is done to find the command. If the path to the command is defined in your shell variable PATH, the full path name is not required.
	args	Arguments to pass to the UNIX command; full path names are required only if the UNIX command requires them.

Options

-f Causes the output of the command to be returned as a result in drawer A. If a file name is entered immediately following the -f option, a copy of the output from the UNIX command is placed into that file. Do not use the -f option when executing an interactive UNIX process such as vi and ed. (Do not use this option with the -w option.)

-p Executes the user's .profile before executing the UNIX command.

-w Allows the run user to press **Resume** to resume the run once the command is executed and the output has been displayed. (Do not use this option with the -f option.)

Outcome

The shell prompt is displayed if you entered the MAPPER system from the UNIX shell. If you entered the MAPPER system using the MAPPER login, a UNIX login prompt is displayed.

Guidelines

Returning to the MAPPER System

To return to the MAPPER system, press **Ctrl-D** or type **exit** and press the **Return** key. The active screen is displayed.

Case Sensitivity

UNIX commands, arguments, and file names are case sensitive. For example, the command /BIN/LS -L does not execute the /bin/ls -l command.

Examples

Example 1

This example displays a long list of the files and directories in the UNIX directory /usr/work/newuser.

```
unix -w /bin/ls -l /usr/work/newuser
```

After pressing **Resume**, the active screen is displayed.

Example 2

This example executes the UNIX cat command on the file prod_status and places a copy of the output in a file named /tempfile:

```
unix -f/tempfile /bin/cat prod_status
```

The output is displayed as a result in drawer A.

UPD (Update)

Use the Update (UPD) function to replace the lines in a report with those displayed in the update result.

Prerequisite

An update result from a previous function must be displayed.

How to Access the Function

Menu Procedure

Procedure	Select menu items using the menu path below.
Menu Path	Select Task Modify Report Extract from a Report

Control Line Procedure

Procedure	Enter one of the following formats on the control line.	
Format	**UPD** [*psw*] Replaces the lines in the original report with those displayed in the update result. The updated report is displayed.	
Format Field	*psw*	Password required to perform an update operation on a write-protected report.

Modifying the Update Result

To modify the lines displayed in the update result before you execute the UPD function, you can do the following:

- Delete lines in the update result that you do not want to use as replacement lines in the original report.
- Add lines to the update result. The added lines are placed at the end of the original report.
- Modify the data in the lines of the update result. The changed lines replace those in the original report.

Functions That Produce Update Results

Calculate Update (CALU)

Change (CHG) with the OU option

Locate (LOC) with the OU option

Match Update (MAU)

Search Update (SU)

VALIDATE

Use the VALIDATE run to verify the existence of data dictionary entries, documentation, and other data dictionary components. You can check the following:

- Data dictionary entries with no documentation
- Runs, screens, and external subroutines that do not contain valid APT tables (see the BAT run in the *Run Design Reference* for details)
- Runs that are not registered in the data dictionary
- Any reports registered in the data dictionary that do not physically exist on the system

Note: *The VALIDATE run is part of the Application Power Tools (APT) feature of MAPPER software. For more information about the VALIDATE run, the data dictionary, and other APT features, see Appendix F.*

How to Access the Run

Menu Access Procedure

Procedure	1. Type **apt** on the control line and transmit. 2. Select **VALIDATE** from the APT menu. 3. Choose a selection from the menu. For help in choosing an item, press the **Help** key. 4. Fill in the subsequent run forms. For help while filling in the forms, press the **Help** key.

Control Line Procedure

Procedure	Enter the following format on the control line.
Format	**VALIDATE** Displays a menu. For help in choosing a selection, press the **Help** key.

For More Information

For more information about the VALIDATE run, enter **apt** on the control line, tab to **VALIDATE**, and press **Help**. For more information about the data dictionary and the runs available with Application Power Tools (APT) feature of MAPPER software, see Appendix F.

Section 5
Iterative Runs

This section applies only to the OS 1100 MAPPER System.

You may find additional information specific to your MAPPER system by pressing **ReadMe** from the sign-on screen.

This section contains the following topics:

- What are iterative runs?
- Using iterative runs
- Iterative run exercise

What Are Iterative Runs?

Each time you use a MAPPER function, you must specify the options and parameters you want to use. Iterative runs are utility runs that correspond to these manual functions and provide you with the ability to enter your options and parameters in a mask, save that mask, and use it again and again. Table 5-1 lists the iterative runs, along with their corresponding manual functions.

Table 5-1. Iterative Runs

Iterative run	Corresponding manual function
IBFN	Binary Find
ICAL	Calculate
ICNT	Count
IDAT	Date
IFND	Find
ISOR	Sort
ISRH	Search
ITOT	Totalize

With the iterative runs, you can use any functionality available with the corresponding manual function, but with greater ease. For example, you can do the following tasks:

- Save completed function masks and use them again.
- Request previously saved masks and alter them without repeatedly filling in entire masks.
- Translate function masks into their run statement formats, which can be easily added to an existing run.

Using Iterative Runs

You can use either the menu procedure or control line procedure to execute an iterative run. For details, see the online topic for the iterative run you are using. For example, for information about the ISOR run, type **help,isor** and transmit.

Outcome

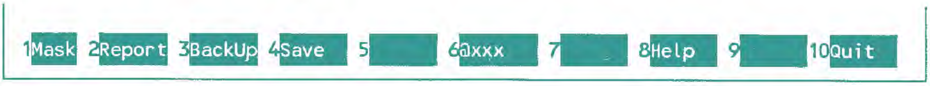
The iterative run performs the corresponding function. For example, if you use the ICAL run, the system performs the computations just as the CAL function does.

If you save the mask, the iterative run places it in the current report or another report you specify. The completed mask appears as a set of asterisk lines at the end of the report. The run updates the report itself; it does not create a result.

If a result was on display when you started the iterative run, the run places the mask at the end of the result. (To save the masks permanently, remember to replace or duplicate the result, creating a permanent report.)

Function Key Bar

After you execute the iterative run and complete the mask, the following function key bar is displayed:



The key actions are as follows:

Mask Redisplays the function mask, allowing you to change the options and parameters and perform the function again.

Report Redisplays the original report or result.

BackUp Displays the previously saved mask. Pressing the **BackUp** key up to three times cycles back through the three most recent masks. Pressing it a fourth time displays the most recent mask.

Save Displays a menu in which you type the name of the mask to be saved and, optionally, the report to save it in. (If you use the name of a previously saved mask, that mask is replaced by the current mask.)

@*xxx* Displays the run statement created from the most recently processed mask. For example, when using the ISOR run, you can press the **@SOR** key to obtain the SOR run statement generated by the options and parameters in the mask.

Saving Masks

You cannot save a mask until after you have performed the corresponding function by transmitting from the function mask.

You can save several masks in the same report, each with a different name. You can include a maximum of six parameter lines in each mask. For example, when using the ICAL run, you may include as many equations as will fit on six lines.

Using the ICAL Run

The ICAL run is particularly useful, since you can build a complex set of equations, one step at a time. There are a few differences between ICAL and the other iterative runs. For example, a report must be on display when you execute the ICAL run and the function forms are slightly different. If you need assistance while filling in a form, press **Help**.

Iterative Run Exercise

To help you understand the iterative runs, try this exercise, which uses the ISOR run.

1. Duplicate report 2B0 and display the duplicate report.
2. Using either the menu procedure or the control line procedure, start the ISOR run.
3. Enter some options and parameters. For ideas, see SORT in Section 4.
4. Transmit to see the sorted result.
5. Press **Mask** to receive the completed mask and change some parameters.
6. Transmit to receive the newly sorted result.
7. Press **Save** to display the Save Mask form.
8. Enter a mask name and transmit.
9. Press **Report** and roll to the end of the report to see the saved mask.

Try using the iterative runs on your own. Remember that if you need help while filling in a mask or form, press **Help**.

Appendix A
Summaries: Functions and Runs

You may find additional information specific to your MAPPER system by pressing **ReadMe** from the sign-on screen.

This appendix contains a complete list of MAPPER manual functions and runs in alphabetical order, and a summary of options for 10 of the most used manual functions.

Functions and Runs

Table A-1 lists MAPPER functions and runs in alphabetical order by name. Optional fields are in brackets ([*xxx*]) and "either or" conditions are in braces ({*xxx* | *xxx*}). A brief description of the function follows the format.

Table A-1. Summary of Functions and Runs

Function/Run Name	Format/Description
Abort	**Abort** key Terminates a function or run in progress.
Acknowledge Message[1]	OK [*response*] Acknowledges the message, removes it from the message queue, and redisplays it on your screen. If the sender requested an acknowledgment, the OK function returns one to the station of any user signed on with the sender's user-id.
Add Line	♦]*q*+[*predfl*] Adds a line or lines.
Add On[1]	ADON ADON *rd*[c] Appends a report to a displayed report or result.
Add Report	AR AR {*rd*[c] | *d*[c]} *title* Adds a new report in the specified drawer.
Add To[1]	ADTO ADTO *rd*[c] Appends the displayed data to another report.
1100: **Alarm**	AL Sends a message for delivery at a specified time. AL,D Displays the message log. AL,DEL Deletes all messages from the log.

continued

[1] Creates a result.

Table A-1. Summary of Functions and Runs (cont.)

<table>
<tr><th>Function/Run Name</th><th>Format/Description</th></tr>
<tr><td><b>Append Line</b></td><td>♦][<i>n</i>]A[<i>b</i>]
Copies lines from a report and appends the lines to an existing temporary buffer.</td></tr>
<tr><td><b>APT Register</b></td><td>APTREG
Registers fields from an existing report or drawer into the data dictionary.</td></tr>
<tr><td><b>APT Select</b></td><td>APTSEL
Selects a default application for APT runs.
APTSEL,<i>appnm</i>
Sets or resets the default to the application specified.
APTSEL,c
Clears a specified application from the default.</td></tr>
<tr><td><b>Arithmetic</b>[1]</td><td>A
A {-|<i>rd</i>[<i>c</i>]}
Performs computations using arithmetic expressions.</td></tr>
<tr><td><b>Auxiliary</b></td><td>AUX
AUX <i>sn</i>[<i>sl rdc f</i>] (not on OS 1100)
Queues data to an auxiliary printer.</td></tr>
<tr><td><b>Background Run</b></td><td>BR <i>run</i>[,<i>d1</i>,<i>d2</i>...]
Executes a run as a background process.
<b>1100:</b> BRX <i>run</i>[,<i>d1</i>,<i>d2</i>...]
Executes a run as a background process and does not send a completion message.</td></tr>
<tr><td><b>Binary Find</b>[2]</td><td>BF
BF {-|<i>rd</i>[c]|<i>d</i>[c]} [<i>f</i>]
Finds and displays data, within sorted reports.</td></tr>
</table>

1. Creates a result.
2. Creates a result with the N or O option.

Table A-1. Summary of Functions and Runs (cont.)

Function/Run Name	Format/Description
Cabinet Switch	C {*cpsw* | *cnm psw*} CS CS [*sl*,]*c*[,*r*] **1100:** CS *c*[,*r*] Selects a cabinet; *cpsw* is the cabinet number and password (for example, C 0open), and *cnm psw* is the cabinet name and password (for example, C home open).
Calculate[1]	CAL CAL *rd*[*c f*] Performs complex computations and conditional evaluations on reports.
Calculate Update[1]	CALU CALU *rd*[*c f*] Performs complex computations and conditional evaluations for update on reports.
Calendar[1]	CALENDAR[,*mmm*,*yyyy*] Displays calendar on screen.
1100: **Call**	CALL CALL *user-id* [*sn*]|*sn* Starts a call session to communicate online with other users.
Change[4]	CHG CHG *rd*[*c f*] or CHG [*rdc f*];/*tgtstr*/*replstr*/[*o*] Changes a character string to a new image.
1100: **Clock**	CLOCK[,*sec*] Displays the current date and a running time clock.
Combine	COMBINE Creates an ad hoc report by combining fields from two or more reports.
Communications Output Printer	COP COP *sn*[,*copies*,*f*,*lsp*] **BTOS:** COP *pntrnm* Sends the current report to the auxiliary printer.

[1] Creates a result.

[4] Creates a result with the O or OU option.

Table A-1. Summary of Functions and Runs (cont.)

Function/Run Name	Format/Description
Compare Data	CMP CMP *rd*[c *f*,*l*] Compares the contents of one report or result with another report.
1100: **Copy**	COPY Copies OS 1100 file or element to another site.
U Series: **Copy Report to DOS**	SAV PCU Copies a MAPPER report to a DOS file in either MAPPER format or straight ASCII format.
1100: **Count**	CNT CNT *rd*[c *f*] Analyzes and summarizes data.
Create File[1]	FILE Copies a report to a native data file.
Create Result Copy[1]	RSLT RSLT *rd*[c] Creates a result copy of a report.
Create Temporary Format	VIEW Changes the selection of report fields displayed.
Date[1]	DATE DATE *rd*[c *f*] Performs computations on dates.
Decode Report[1]	DECODE **1100:** DECODE *key* Transforms encoded report into readable report.

[1] Creates a result.

Table A-1. Summary of Functions and Runs (cont.)

Function/Run Name	Format/Description
Delete[3]	DEL [*psw*] Deletes lines of an update result from a report.
Delete Line	♦]*q*- Deletes a line or lines.
Delete Report	DR DR *rd*[*c*] Deletes a report.
Delete Share	DELSHR Removes the link between applications that share specific data dictionary information.
Device[1]	DEV DEV,*sn* (not on BTOS) **1100:** DEV,*sn*[*dev*] Lists auxiliary devices attached to a terminal.
1100: **Diskette**	DISK Reads from or writes to 5 1/4 inch diskettes.
Display Alternative Format	F*n* Displays a report in a different format.
Display and Hold Headings	DH Holds the headings at the top of the screen as you roll through the report.
Display Line Count	DLC[,S] Places line numbers of report or result on screen display.
Display Report	D *rd*[*c f*,*l*] or D *datanm* [*f*,*l*] Displays a report.

1. Creates a result.
3. Use with update functions:
   - Calculate Update
   - Change (with the OU option)
   - Locate (with the OU option)
   - Match Update
   - Search Update

Table A-1. Summary of Functions and Runs (cont.)

Function/Run Name	Format/Description
U Series: **DOS**	DOS PCU Enters DOS.
1100: **Downline Load**	DLL Downline loads UTS 400/40 terminal.
Drawer Password	DPW Enter Drawer Password The message displayed when you try to access a report that has a drawer password.
Drawer Table of Contents	T Displays the available drawers in your cabinet.
Duplicate Line	♦]xX[*q*] Duplicates a line or lines.
Duplicate Report	XR XR {*rd*[c] | *d*[c]} [*title*] Duplicates a report.
1100: **Element**	ELT Copies MAPPER reports to an OS 1100 file or element.
1100: **Element Delete**	ELT- Deletes an OS 1100 file or element.
Encode Report[1]	ENCODE **1100:** ENCODE *key* Transforms report into unreadable code.
Entry	ENTRY ENTRY,*entry-type* Adds, changes, or deletes items in the data dictionary.
Exit MAPPER System	EXIT Exits MAPPER system and returns to previous environment.

[1] Creates a result.

Table A-1. **Summary of Functions and Runs** (cont.)

Function/Run Name	Format/Description
Extract[3]	EXT [*psw*] Deletes lines of an update result from the original report and redisplays the result.
Find	F F {- | *rd*[c] | *d*[c]} [*f*] Finds and displays data.
Form Generation	FORMGEN Builds report for generating new drawer.
Form Generation (APT)	FORMG Creates an experimental report.
U Series: **Harvard Graphics**	HG PCU Executes Harvard Graphics software and, optionally, sends a MAPPER report for processing.
Help	HELP HELP,*topic* Provides information on MAPPER capabilities and usage.
1100: **Hold Characters on Screen**	HC*n* Holds a specified number of characters on the left side of the screen as you shift the report.
Hold Lines on Screen	H*n* Holds a specified number of lines at the top of the screen as you roll through the report.
Index[1]	I I [*q*]*d* Indexes a drawer.

1. Creates a result.
3. Use with update functions:
   - Calculate Update
   - Change (with the OU option)
   - Locate (with the OU option)
   - Match Update
   - Search Update.

Table A-1. Summary of Functions and Runs (cont.)

Function/Run Name	Format/Description
Index User[1]	IU IU [*q*]*d* Indexes a drawer by user, update dates, and range of reports.
Information	INFO INFO,*entry-type,appnm* Displays a result containing the data dictionary information for one or more items.
Insert Line	♦]*xl/*[,*q* | *-l/*] Inserts a line or lines.
1100: **Iterative Binary Find**	IBFN[*] IBFN[*] {*rd* | *d*}[,*f*,*masknm*,*maskrep*] Generates Binary Find function masks that you can save, alter, and use again.
Iterative Calculate	ICAL[*] ICAL[* *eqnm*,*r*] Creates Calculate equation sets that can be saved, altered, and used again.
1100: **Iterative Count**	ICNT[*] ICNT[*] {*rd* | *d*}[,*f*,*masknm*,*maskrep*] Generates Count function masks that you can save, alter, and use again.
1100: **Iterative Date**	IDAT[*] IDAT[*] *rd*[,*f*,*masknm*,*maskrep*] Generates Date function masks that you can save, alter, and use again.
1100: **Iterative Find**	IFND[*] IFND[*] {*rd* | *d*}[,*f*,*masknm*,*maskrep*] Generates Find function masks that you can save, alter, and use again.

1. Creates a result.
4. Creates a result with the O or OU option.

Table A-1. Summary of Functions and Runs (cont.)

Function/Run Name	Format/Description
1100: **Iterative Search**	ISRH[*] ISRH[*] {*rd*|*d*}[,*f*,*masknm*,*maskrep*] Generates Search function masks that you can save, alter, and use again.
1100: **Iterative Sort**	ISOR[*] ISOR[*] *rd*[,*f*,*masknm*,*maskrep*] Generates Sort function masks that you can save, alter, and use again.
1100: **Iterative Totalize**	ITOT[*] ITOT[*] *rd*[,*f*,*masknm*,*maskrep*] Generates Totalize function masks that you can save, alter, and use again.
Kill	KILL KILL {*run*|,*sn*|*run sn*} Terminates an active run.
Language	LANG LANG *n* Specifies the language to display system messages.
Line Control	L Restores the control line when it is overwritten.
Line Zero	LZ LZ *rd*[c] Displays identification information.
Lists	LISTS Displays lists of specific information from the data dictionary.
Locate[4]	LOC LOC *tgtstr* or LOC *rd*[c *f*] LOC [*rdc f*];/*tgtstr*/[o] Locates and displays a character string.

1. Creates a result.

Table A-1. Summary of Functions and Runs (cont.)

Function/Run Name	Format/Description
Look Switch	LOOKSW Alternates between the two user interfaces of MAPPER software — menu interface and control line interface.
U Series: **Lotus 1-2-3**	LOTUS PCU Executes Lotus 1-2-3 software and, optionally, sends a MAPPER report for processing.
Match[1]	MA MA *rd*[*c f*] Compares, matches, and moves data.
Match Update[1]	MAU MAU *rd*[*c f*] Matches data for update.
Message Waiting	MSG Displays a message sent to your station or user-id.
U Series: **Microsoft Word**	WORD PCU Executes Microsoft Word software and, optionally, sends a MAPPER report for processing.
Move Line	♦]xM/[,*q* | -*//*] Moves a line or lines.
Name	NAME NAME *nm*[*,c,d,r,dept,user-id,act,desc*] **1100:** NAME *nm*[*,c,d,r,dept,user-id,func,upd?*] Creates a directory name for a cabinet, drawer, or report.
Names[1]	NAMES Displays data names registered in system directory.
Paint	PNT Restores the screen.

[1] Creates a result.

Table A-1. Summary of Functions and Runs (cont.)

Function/Run Name	Format/Description
Pass	PASS Shares data dictionary information with users of different applications or with the public dictionary.
Password	PSW *psw* Assigns, changes, or clears report write passwords; unlocks reports for updating.
Print	PR PR *rd*[c *f*] (not on OS 1100) Queues data to a system printer.
Pull	PULL Accesses information in the public dictionary.
Put Line	♦]P[*b*] Copies lines from the temporary yank or append buffers into a report or result.
Read Password	RPSW RPSW {*psw* | *keyw*} Limits access to individual reports.
Reformat Report[1]	RF RF *rd*[c *f*] Moves columns of data.
Release Display	^ Releases displays or runs.
Remote Run	RR *run*[,*d1*,*d2*...] (not on BTOS or OS 1100) Starts a MAPPER run at a remote site.
1100: **Remote Symbiont Interface**	RSI Enters demand mode at MAPPER terminal.
Replace Report	REP REP *rd*[c *title*] Replaces a report with a displayed report or result.

[1] Creates a result.

Table A-1. Summary of Functions and Runs (cont.)

Function/Run Name	Format/Description
Report and Line Limits	LIMITS Displays the highest report number allowed and the maximum number of lines per report allowed for cabinet and drawer.
Resume	RSM Resumes executing a suspended function or run.
Retrieve File[1]	RET Retrieves a native data file. **1100:** RET [P] Retrieves an OS 1100 data file or program file element.
1100: **Retrieve Report**[1]	RETR Retrieves a MAPPER report.
1100: **Run Status**	RS RS[,*o*,*run*,*sn*] Displays the status of runs.
Save Report Version[1]	SV Renames a report or result for temporary storage. After doing this, press **Resume** or enter **rsm** (the Resume function) to redisplay it as a result.
1100: **Schedule**	SCHEDULE SCHEDULE *run*,*time*,*date* Schedules a run for execution at a later time or for periodic execution. SCHEDULE D Displays information about a currently scheduled run.
Search[1]	S S {-|*rd*[c]|*d*[c]} [*f*] Searches for data.
Search Update[1]	SU SU {-|*rd*[c]|*d*[c]} [*f*] Searches reports for update.

[1] Creates a result.

Table A-1. Summary of Functions and Runs (cont.)

Function/Run Name	Format/Description
Send Report	SEND SEND *sn*[*sl,ack?*] **1100:** SEND *sn*[*,ack?*] Sends a report or result to another station.
Send Report to User	SNU SNU *user-id*[*,dept,sl,ack?*] **1100:** SNU *user-id*[*,dept,ack?*] Sends an entire report or result to another user.
Shift Display	S*n* Shifts the displayed portion of the report.
Sign off MAPPER Software	X Signs station off from the MAPPER software.
Sign on MAPPER Software	♦]*user-id,dept*[*,psw*] Signs on to MAPPER software.
1100: **Snooze**	SNOOZE SNOOZE *time* Delays receipt of messages.
SOE Update	♦*changes*█ Changes data between SOE (♦) and the cursor (█).
Sort[1]	SORT SORT *rd*[c *f*] Reorders lines of data.
Sort and Replace Report	SORTR SORTR *rd*[c *f*] Reorders the lines in a report and replaces the sorted result into the original report.
Start	START Starts a job in the native environment.
Station-to-Station Message[1]	SS[,C] Sends messages (up to a full screen) to another station.

[1] Creates a result.

Table A-1. Summary of Functions and Runs (cont.)

Function/Run Name	Format/Description
1100: **Stop**	STOP STOP,*run*[,*sn*] Terminates background runs.
System[1]	SYSTEM (not BTOS or OS 1100) Displays active users, active runs, and other system information.
1100: **Tape Cassette**	TCS Reads from or writes to cassettes or diskettes.
1100: **Terminal Width**	TW {80|132} Sets display width to 80 or 132 columns.
Totalize[1]	TOT TOT *rd*[*c f*] Performs arithmetic and move operations.
U Series: **Transfer DOS File to MAPPER**	RES PCU Transfer a DOS file to the MAPPER system.
Undo	UNDO Reverses the last operation (only available for some functions).
U Series: **UNIX Interface**[5]	UNIX [*o*] [*cmd* [*arg*]] Accesses the UNIX system and lets you issue UNIX commands.

1. Creates a result.
5. Creates a result with the -f option.

Table A-1. Summary of Functions and Runs (cont.)

Function/Run Name	Format/Description
Update[3]	UPD [*psw*] Blends lines in the results of an update function into reports.
Validate	VALIDATE Verifies the existence of data dictionary entries, documentation, and other components.
Yank Line	♦][*n*]Y[*b*] Copies lines from a report or result into a temporary buffer.

[3] Use with update functions:

- Calculate Update
- Change (with the OU option)
- Locate (with the OU option)
- Match Update
- Search Update

Abbreviations for Functions and Runs

ack?	Acknowledgement request? Y/N.
act	Action (C,D,I,R,or,Q)
appnm	Application name
arg	Arguments
b	Buffer number
c	Cabinet number
changes	Modifications to report
cmd	Command
cnm	Cabinet name
copies	Number of copies
cpsw	Cabinet number and password
d	Drawer letter
d1,d2...	Data items
datanm	Report name from system directory
date	Date
dept	Department number
desc	Description of named cabinet, drawer, or report
dev	Device type
entry-type	Data dictionary entry types (F, C, R, E, S, M, A)
eqnm	Equation set name
f	Format
func	Action (ADD, CHG, DEL)
id	Identifier name
key	Character key
keyw	Special keyword
l	Line number
ll	Last line number
lsp	Line spacing
masknm	Name of existing or new function mask
maskrep	Report containing previously saved function mask
mmm	First three letters of month
msg	Message
n	Number
nm	Name
o	Options
predfl	Predefined lines defined in report 0
pntrnm	Printer name
psw	Password
q	Quantity of lines
qd	Quantity of lines to index and drawer
r	Report number
rd	Report and drawer
rdc	Report, drawer, and cabinet
replstr	Replacement string
response	Response
run	Run name
sec	Number of seconds (1 through 60)
sl	Site letter
sn	Station number

tgt	Target
tgtstr	Target string
time	Time in minutes (default = 60), hours (H), or days (D; up to one week)
title	Title of the report
topic	Code on main help menu
upd?	Update names in system directory? Y/N
user-id	User-id
x	Number of times to duplicate a line
yyyy	Year
-	Report or result on display
*	Use report field names

Options for 10 Common Functions

The following list includes the options for 10 common manual functions:

	BF	A, B, C[2], E, Fn, I[n], K, N, O, P, Q, Rx{-y}[4], S, U, @, /
	CAL	A, C, E, I, J(x), K[n], L, N[n], O, Rn, S(s), T, V, X, *
▽	1100:	S(x) instead of S(s)
	CHG	A, C[2], F, M[3], O, OU, Sx{-y|,n}, Tx
	DATE	A, T, W, n
	F	A, C[2], Rx{-y|y}, @, /
	LOC	A, B[n][1], C[2], F, M[3], O, OU, Sx{-y|,n}, Tx, U[1]
	MA	A, B, C[2], D, E, F, I[1], M, N, P, Q, S
	S	A, B[(n)], C[2], D, F, H, L(x), N, P, Q[(n)], Rx{-y|y}, T[(x)], U[(x)], @, /
▽	1100:	Also En and S
	SORT	A, C[2]
▽	1100:	Also X+ and X-
	TOT	A, C[2], E, H, I, J(x), N, O, Rn, S, =x, *

1. Not applicable in MAPPER runs.
2. Option C varies with function.
3. LCH = Change manual function; M option not applicable for manual Change and Locate functions.
4. For BF (Binary Find), only Rx and Rx-y are allowed.

Table A-2 summarizes the options for 10 commonly used manual functions.

Table A-2. Options for 10 Common Manual Functions

Option	Purpose	Functions				
A	Process all line types.	BF	CAL	CHG	DATE	F
		LOC	MA	SORT	S	TOT
B	Build index.	BF				
B[*n*][1]	Back up *n* lines after locate.	LOC				
B	Blend issuing and receiving reports.	MA				
B[(*n*)]	Stop search after *n*th find. Default = first find.	S				
C[2]	Case sensitivity (character se control on 1100).	BF	CHG	F	LOC	
		MA	SORT	S	TOT	
C	Conditionally display specific result lines.	CAL				
D	Omit match or search information lines from result.	MA	S			
E	Display last occurrence of item found.	BF				
E	Do not move blank fields from issuing report.	MA				
E	Erase fields (fill with spaces) if value = 0.	CAL				
E	Count entries.	TOT				
1100: E*n*	Estimate number of lines in result.	S				
F	Process all line types; locate or change full character string.	CHG	LOC			
F	Do not fill move fields on a no-match condition.	MA				
F	Search for floating-point numbers.	S				
F*n*,*n*...	Specify order of sorted fields.	BF				

continued

1. Not applicable in MAPPER runs.
2. Option C varies with function.
3. LCH = Change manual function; M option not applicable for manual Change and Locate functions.
4. For BF (Binary Find), only R*x* and R*x*-*y* are allowed.

Table A-2. Options for 10 Common Manual Functions (cont.)

Option	Purpose	Functions	
H	Display only heading lines from first report in multiple report search.	S	
H	Cumulate horizontally.	TOT	
I[*n*]	Use index in report *n*. Default = report 2.	BF	
I	Produce integer results.	CAL	
I[1]	Issuing report on display.	MA	
I	Ignore heading restrictions.	TOT	
J(*x*)	Numerically justify result value to *x*: x = c, l, r, x, or z.	CAL	TOT
K	Verify that reports are sorted in ascending order.	BF	
K[*n*]	Initialize value label to *n*.	CAL	
L	List value label names or values in result.	CAL	
L(*x*)	Omit line type *x* from result.	S	
M[3]	Treat first character of target string as line type designator.	CHG	LOC
M	Display only matched lines in result.	MA	
N	Create separate line per item and item count in result.	BF	
N	Display three blank lines between subtotal groups and omit grand total.	TOT	
N[*n*]	Substitute numeric value *n* for nonnumeric fields. Default = 0.	CAL	
N	Display lines not meeting match or search parameters.	MA	S

1. Not applicable in MAPPER runs.
2. Option C varies with function.
3. LCH = Change manual function; M option not applicable for manual Change and Locate functions.
4. For BF (Binary Find), only R*x* and R*x*-*y* are allowed.

Table A-2. Options for 10 Common Manual Functions (cont.)

Option	Purpose	Functions		
O	Create result containing items found.	BF	CHG	LOC
O	Omit data lines from result; include headings, value labels or totals.	CAL	TOT	
OU	Create update result.	CHG	LOC	
P	Include all types of lines in result (valid only with N option).	BF		
P	Issuing and receiving reports are presorted.	MA		
P	Process paragraphs.	S		
Q	Find a single item quickly.	BF		
Q	Specifies reports are sorted and will not be verified.	MA		
Q[(*n*)]	Stop scan after *n*th paragraph. Default = first paragraph; use with the P option.	S		
R*x*{-*y*|,*y*}[4]	Scan range of reports: reports *x* through *y*; scan reports *x*,*y*.	BF	F	S
R*n*	Round answers to nearest *n*.	CAL	TOT	
S	Scan each report separately.	BF		
S*x*{-*y*|,*n*}	Start scan at line *x* through line *y*; scan *n* lines.	CHG	LOC	
S	Display matched lines in issuing report order.	MA		
1100: S	Display found lines in search parameter order.	S		
S	Place subtotals in vertical operation fields.	TOT		
S(s)	Case sensitivity.	CAL		
1100: S(*x*)	Define character set interpretation. *x* = f, l, or s.	CAL		

1. Not applicable in MAPPER runs.
2. Option C varies with function.
3. LCH = Change manual function; M option not applicable for manual Change and Locate functions.
4. For BF (Binary Find), only R*x* and R*x*-*y* are allowed.

Table A-2. Options for 10 Common Manual Functions (cont.)

Option	Purpose	Functions		
T	Include processed and unprocessed lines in result.	CAL		
T*x*	Set *x* to transparent character.	CHG	LOC	
T	Convert time in field to decimal hours and move field.	DATE		
T[(*x*)]	Include last *x* type line in result. Default = tab line.	S		
U	Set update lock.	BF		
U[1]	Resume scan beyond lines on display.	LOC		
U[(*x*)]	Search within data unit; include unit in result. Default = tab line.	S		
V	Process only equations whose result values are calculated from valid data.	CAL		
W	Convert dates into days of the week.	DATE		
X	Exclude invalid values in MIN, MAX, SUM, AVG, VMIN, VMAX, VSUM, and VAVG computations.	CAL		
1100: X+	Use OS 1100 Sort/Merge regardless of report length.	SORT		
1100: X-	Use normal MAPPER software sorting process regardless of report length.	SORT		
n	Specify *n* workdays in week. Default = 7.	DATE		
@	Find or search for blank characters (spaces).	BF	F	S
/	Find or search for slant as data.	BF	F	S
=*x*	Change column 1 to *x*.	TOT		
*	Omit error flag (*) in subtotaling operations.	TOT		
*	Flag invalid results with asterisk.	CAL		

1. Not applicable in MAPPER runs.
2. Option C varies with function.
3. LCH = Change manual function; M option not applicable for manual Change and Locate functions.
4. For BF (Binary Find), only R*x* and R*x*-*y* are allowed.

Appendix B
Character Sets and Sorting Orders

You may find additional information specific to your MAPPER system by pressing **ReadMe** from the sign-on screen.

This appendix contains the following topics:

- ASCII character set
- Fieldata (limited character set)
- Using the C(S) option
- Character set processing on the OS 1100 MAPPER System

ASCII Character Set

Table B-1. ASCII Character Set

Character	ASCII Internal Octal Code	ASCII Decimal Code	Hexadecimal Code
Tab	11	9	9
Space	40	32	20
!	41	33	21
"	42	34	22
#	43	35	23
$	44	36	24
%	45	37	25
&	46	38	26
'	47	39	27
(	50	40	28
)	51	41	29
*	52	42	2A
+	53	43	2B
,	54	44	2C
-	55	45	2D
.	56	46	2E
/	57	47	2F
0	60	48	30
1	61	49	31
2	62	50	32
3	63	51	33
4	64	52	34
5	65	53	35
6	66	54	36
7	67	55	37
8	70	56	38
9	71	57	39
:	72	58	3A
;	73	59	3B
<	74	60	3C
=	75	61	3D
>	76	62	3E
?	77	63	3F
@	100	64	40
A	101	65	41

continued

Table B-1. ASCII Character Set (cont.)

Character	ASCII Internal Octal Code	ASCII Decimal Code	Hexadecimal Code
B	102	66	42
C	103	67	43
D	104	68	44
E	105	69	45
F	106	70	46
G	107	71	47
H	110	72	48
I	111	73	49
J	112	74	4A
K	113	75	4B
L	114	76	4C
M	115	77	4D
N	116	78	4E
O	117	79	4F
P	120	80	50
Q	121	81	51
R	122	82	52
S	123	83	53
T	124	84	54
U	125	85	55
V	126	86	56
W	127	87	57
X	130	88	58
Y	131	89	59
Z	132	90	5A
[	133	91	5B
\	134	92	5C
]	135	93	5D
^	136	94	5E
_	137	95	5F
‘ (grave accent)	140	96	60
a	141	97	61
b	142	98	62
c	143	99	63

Table B-1. ASCII Character Set (cont.)

Character	ASCII Internal Octal Code	ASCII Decimal Code	Hexadecimal Code
d	144	100	64
e	145	101	65
f	146	102	66
g	147	103	67
h	150	104	68
i	151	105	69
j	152	106	6A
k	153	107	6B
l	154	108	6C
m	155	109	6D
n	156	110	6E
o	157	111	6F
p	160	112	70
q	161	113	71
r	162	114	72
s	163	115	73
t	164	116	74
u	165	117	75
v	166	118	76
w	167	119	77
x	170	120	78
y	171	121	79
z	172	122	7A
{	173	123	7B
|	174	124	7C
}	175	125	7D
~	176	126	7E

Fieldata (Limited Character Set)

This section applies only to the OS 1100 MAPPER System.

Table B-2. Limited Character Set

Character	Fieldata Internal Octal Code
@	00
space	05
A	06
B	07
C	10
D	11
E	12
F	13
G	14
H	15
I	16
J	17
K	20
L	21
M	22
N	23
O	24
P	25
Q	26
R	27
S	30
T	31
U	32
V	33
W	34
X	35
Y	36
Z	37
)	40
-	41
+	42
<	43
=	44
>	45
&	46

continued

Table B-2. Limited Character Set (cont.)

Character	Fieldata Internal Octal Code
$	47
*	50
(	51
%	52
:	53
?	54
!	55
, (comma)	56
\	57
0	60
1	61
2	62
3	63
4	64
5	65
6	66
7	67
8	70
9	71
' (apostrophe)	72
;	73
/	74
. (period)	75

Using the C(S) Option

1100: For the OS 1100 MAPPER System, the following information applies only to FCS reports.

Some functions allow you to specify case sensitivity by using the C(S) option. When you use a function without the C(S) option, the system treats lowercase and uppercase characters the same. For example, when you sort a report, the letters **a** and **A** appear together in sort order.

When you use the C(S) option, the report is sorted in the strict order of the ASCII codes: uppercase with uppercase and lowercase with lowercase.

These differences also affect a range search. For example, a range search from **a** to **z** with no options produces **a** through **z** and **A** through **Z**. In the same range search with the C(S) option, the uppercase letters **A** through **Z** are not included in the result.

Table B-3 shows the sort order of the characters with the C(S) option. Table B-4 shows the order without the C(S) option. Both tables start at the top of the leftmost column and go down, then from the top of the second column and so on.

1100: Tables B-3 and B-4 do not apply to the OS 1100 MAPPER System; the sort order for LCS is shown in Table B-5 and the sort order for FCS is shown in Table B-6.

Table B-3. Sorting with the C(S) Option

Tab	-	;	I	W	d	r
Space	.	<	J	X	e	s
!	/	=	K	Y	f	t
"	0	>	L	Z	g	u
#	1	?	M	[	h	v
$	2	@	N	\	i	w
%	3	A	O	]	j	x
&	4	B	P	^	k	y
'	5	C	Q	_	l	z
(	6	D	R	`(grave	m	{
)	7	E	S	accent)	n	|
*	8	F	T	a	o	}
+	9	G	U	b	p	~
,	:	H	V	c	q	

Table B-4. Sorting without the C(S) Option

Tab	-	;	e	l	s	z
Space	.	<	E	L	S	Z
!	/	=	f	m	t	[
"	0	>	F	M	T	\
#	1	?	g	n	u	]
$	2	@	G	N	U	^
%	3	a	h	o	v	_
&	4	A	H	O	V	`(grave
'	5	b	i	p	w	accent)
(	6	B	I	P	W	{
\	7	c	j	q	x	|
*	8	C	J	Q	X	}
+	9	d	k	r	y	~
,	:	D	K	R	Y	

Character Set Processing on the OS 1100 MAPPER System

▽ *This section applies only to the OS 1100 MAPPER System.*

The OS 1100 MAPPER System processes Fieldata and ASCII character sets. You cannot use certain characters as data (namely, those characters used as control characters in report processing).

LCS, FCS, and FCSU

The Fieldata character set used in MAPPER software is called limited character set (LCS); the ASCII character set is called full character set (FCS). Also, you can store and process FCS report data in uppercase alphabetic characters only — called full character set upper (FCSU).

The character set of a drawer is determined when the drawer is generated. To see which character set a report is in, use the Line Zero (LZ) function.

Character Hierarchy

The hierarchy (that is, the relationship of characters to one another) is different between the LCS and FCS character sets. Since MAPPER software actually processes the data as represented by the Fieldata and ASCII codes, the order of the Fieldata or ASCII codes determines the character hierarchy.

For example, in LCS, numeric characters have higher values than alphabetic characters. Therefore, in sort or search-in-range processes, alphabetic characters come before numeric characters. On the other hand, in the FCS character set, alphabetic characters have higher values than numeric characters.

Consider these intercharacter relationships carefully, especially when processing drawers that use different character sets.

Using the C(x) Option

Under normal report or result processing, the functions assume that you want to process the characters according to their native code order.

With some functions, you can use the C(*x*) option to process data with intercharacter relationships different from their native code order. You can use one of the following:

C(F) Process using full character set (ASCII) order
C(L) Process using limited character set (Fieldata) order
C(S) Process using the strict character order; that is, treat uppercase and lowercase characters differently

Using the C(L) and C(F) Options

When using a function that processes more than one report, and the reports are in different character sets, you may need to use the C(F) or C(L) option to indicate which character set order to use for the function. This ensures that the reports use the same hierarchy.

For example, suppose you want to match two reports in different character sets. You plan to sort the reports first and use the P option of the Match function, which specifies that the reports are presorted.

If the receiving report is in LCS and the issuing report is in FCS, use the C(L) option when sorting the issuing report.

Conversely, if the issuing report is in LCS and the receiving report is in FCS, use the C(F) option when sorting the issuing report.

Table B-5 shows the LCS order when using the C(*x*) option. Table B-6 shows the FCS order when using the C(*x*) option.

Table B-5. LCS Order Using the C(x) Option

Basic or Sorted C(L) or C(S)	Sorted FCS C(S)	Sorted FCS C(F)
@	@	Tab
Tab	Tab	Space
Space	Space	!
A	A	$
B	B	%
C	C	&
D	D	'
E	E	(
F	F	)
G	G	*
H	H	+
I	I	,
J	J	-
K	K	.
L	L	/
M	M	0
N	N	1
O	O	2
P	P	3
Q	Q	4
R	R	5
S	S	6
T	T	7
U	U	8
V	V	9
W	W	:
X	X	;
Y	Y	<
Z	Z	=
)	)	>
-	-	?
+	+	@
<	<	A
=	=	B
>	>	C

continued

Table B-5. LCS Order Using the C(x) Option (cont.)

Basic or Sorted C(L) or C(S)	Sorted FCS C(S)	Sorted FCS C(F)
&	&	D
$	$	E
*	*	F
(	(	G
%	%	H
:	:	I
?	?	J
!	!	K
'	'	L
\	\	M
0	0	N
1	1	O
2	2	P
3	3	Q
4	4	R
5	5	S
6	6	T
7	7	U
8	8	V
9	9	W
,	,	X
;	;	Y
/	/	Z
.	.	\

Table B-6. FCS Order Using the C(x) Option

Basic or Sorted	Sorted Sorted	
C(F)	C(S)	C(L)
Tab	Tab	@
Space	Space	Tab
!	!	[
"	"	]
#	#	#
$	$	Space
%	%	A,a
&	&	B,b
'	'	C,c
(	(	D,d
)	)	E,e
*	*	F,f
+	+	G,g
,	,	H,h
-	-	I,i
.	.	J,j
/	/	K,k
0	0	L,l
1	1	M,m
2	2	N,n
3	3	O,o
4	4	P,p
5	5	Q,q
6	6	R,r
7	7	S,s
8	8	T,t
9	9	U,u
:	:	V,v
;	;	W,w
<	<	X,x
=	=	Y,y
>	>	Z,z
?	?	)
@	@	-
A,a	A	+

continued

Table B-6. FCS Order Using the C(x) Option (cont.)

Basic or Sorted C(F)	Sorted Sorted C(S)	C(L)
B,b	B	<
C,c	C	=
D,d	D	>
E,e	E	&
F,f	F	$
G,g	G	*
H,h	H	(
I,i	I	%
J,j	J	:
K,k	K	?
L,l	L	!
M,m	M	,
N,n	N	\
O,o	O	0
P,p	P	1
Q,q	Q	2
R,r	R	3
S,s	S	4
T,t	T	5
U,u	U	6
V,v	V	7
W,w	W	8
X,x	X	9
Y,y	Y	'
Z,z	Z	;
[	[	\
\	\	:
]	"	"
—	-	—
	a	
{	b	{
|	c	|
}	d	}
~	e	~
	f	

Table B-6. FCS Order Using the C(x) Option (cont.)

Basic or Sorted	Sorted Sorted	
C(F)	C(S)	C(L)
	g	
	h	
	i	
	j	
	k	
	l	
	m	
	n	
	o	
	p	
	q	
	r	
	s	
	t	
	u	
	v	
	w	
	x	
	y	
	z	
	{	
	|	
	}	
	~	

Appendix C
Using Your Keyboard and Mouse

You may find additional information specific to your MAPPER system by pressing **ReadMe** from the sign-on screen.

This appendix describes keys available on the function key bar and the use of the MAPPER software mouse.

Function Keys

The keys referred to in Table C-1 correspond to the keys available on the function key bar. Since your needs change depending on your task at the time, the keys available on the function key bar change to fit your task. For example, since you can edit a report, the **Edit** key appears in the function key bar when you have a report on display. Since you cannot edit a function form, the **Edit** key does not appear in the function key bar when a function form is on display.

Table C-1. Function Key Descriptions

Function Key	Description
AddLin	Adds a line to the report where the cursor is positioned.
Edit	Brings you into edit mode. From there you can update a report without using an SOE character. Also from there you have access to the Line Change menu from which you can choose to add, delete, or duplicate lines.
Exit	Takes you out of the MAPPER system and back into the local environment.
Format	Lists the formats and procedures to access a given function through the control line. This key appears only when a help screen for a specific topic is on display. For example, if help for the Totalize function is on display, press **Format** to display a list of formats available for use with the control line procedure.
Help	Displays context-sensitive help and general-system help. For example, if you need help while filling out a function form, press **Help** and specific instructions are displayed. Note that the **Help** key is always available.
LineCh	Displays a menu from which you can choose how to change a certain line. For example, to delete a line, place the cursor on the line to be deleted and press **LineCh**. Then move the cursor to the **Delete line** selection and transmit. This function key appears on the function key bar only after the **Edit** key has been pressed.
KeyHlp	Lists the names of some of the keys used in the documentation and their corresponding keys on your keyboard. For example, the text in **KeyHlp** might explain that the **Abort** key in the documentation is equivalent to a specific key on your keyboard. If you need definitions for keys other than those listed, refer to the printed documentation for your terminal.

continued

Table C-1. Function Key Descriptions (cont.)

Function Key	Description
Menus	Describes how to access a certain function through the menus. This key appears only when a help screen for a specific topic is on display. For example, if help for the Totalize function is on display, press **Menus** to display the menu path and procedure to access the Totalize function.
Op&Prm	Lists options and parameters for a given function. This key appears only when a help screen for a specific topic is on display. For example, if help for Totalize is on display, press **Op&Prm** to get a list of Totalize options and parameters.
Paint	Repaints the screen. For example, with a report on display type some text but do not transmit. Now if you press **Paint**, the report is redisplayed without the text you just typed.
Quit	Displays the active screen. If a help screen is on display, this key redisplays the screen you were viewing before you requested help.
ReadMe	Describes information, if any, specific to your MAPPER system. This information may not be documented anywhere else.
Relatd	Lists the screen titles of the current help topic and of topics related to the current help topic. This key applies only when a help screen is on display. For example, if the "Computing Data Using Calculate" help screen is displayed, press **Relatd** to find other help topics related to CAL, such as "Computing Data Using Arithmetic" or "Computing Data Using Totalize."
Remote	Lists systems to which you have access. You can transfer your station to another MAPPER site by pressing the **Remote** key, moving the cursor to a system on the menu, and transmitting.
Report	Lists all the reports to which you have access. You can choose a report in which to work by pressing **Report**, moving the cursor to a report on the menu, and transmitting.
Resume	Continues a function from the point at which it pauses or stops. For example, since the Find function pauses after it finds the first occurrence of a character string, you must press **Resume** to find successive occurrences.
Return	Displays a previous screen. For example, if you display a report while the active screen is on display, pressing **Return** then redisplays the active screen. (The screen you are returned to varies depending upon the current screen display and the path you used to obtain it.)

Table C-1. Function Key Descriptions (cont.)

Function Key	Description
RollFw	Moves to the next screen in a sequence of screens.
RollBk	Moves to the previous screen in a sequence of screens.
Runs	Lists all the runs to which you have access. You can execute one of the runs by pressing **Runs**, moving the cursor to one of the run names, and transmitting.
SgnOff	Brings you back to the sign-on screen. From there, you can sign on to MAPPER again or exit the MAPPER system.
SignOn	Signs you on to the MAPPER system as a new user. This gives you access to the demonstration database and the basic manual functions. This should be used for training purposes only.
SOE	Places an Start-of-Entry (SOE) character at the cursor position for data updates. ▽ **1100:** The **SOE** function key is not available on the OS 1100 MAPPER System. Use the SOE key defined in the documentation for your terminal.
Tasks	Displays a selection of jobs you might wish to accomplish, such as printing, comparing reports, or finding data. For example, if you want to print a report, press **Tasks**, then select **Print** from that menu. Another menu then displays specific print functions from which you can choose.
Undo	Reverses the last update function performed. This key does not undo the action of a run, but it does reverse the action of some functions, for example, deleting a report. Note that you can only undo the last operation if the **Undo** key appears on the function key bar.
View	Displays a menu of selections that allow you to control the report display. For example, to shift the report 35 characters to the left, press **View**, then enter 35 in the **Shift** menu selection and transmit.

Table C-2 indicates which function keys appear on certain MAPPER screens. For example, note that the **Help** key is available on every screen. The **LineCh** key, however, is available only while the system is in the report editing mode.

1100: The **SOE** function key is not available on the OS 1100 MAPPER System. Use the SOE key defined in the documentation for your terminal.

Note: *The underlining in this table indicates that the key is not available unless the action is possible. For example, after you have performed a function that can be undone, the* ***Undo*** *key is displayed.*

Table C-2. Screen Location of Keys on the Function Key Bar

	...these keys are available									
When this screen is on display...	1	2	3	4	5	6	7	8	9	10
Sign-on	SignOn	KeyHlp				ReadMe		Help		Exit
Active	Report	KeyHlp	Runs			Tasks	Remote	Help		SgnOff
Drawer Select menu		RollFw	RollBk	Return		Tasks		Help		Quit
Report Select menu		RollFw	RollBk	Return		Tasks		Help		Quit
Select Run menu		RollFw	RollBk	Return				Help		Quit
Select System menu (Remote Selection)		RollFw	RollBk	Return				Help		Quit
Report	Resume	Paint	SOE	Return		Tasks	View	Help	Undo	Edit
Report Edit	AddLin	Paint	SOE	Return		LineCh	View	Help	Undo	
Line Change menu	Resume	Paint		Return				Help		Quit
Select Task menu										
(Active screen)				Return				Help		Quit
(Report screen)		Paint		Return				Help		Quit
Change View menu	Resume	Paint		Return				Help		Quit
Function form	Resume	Paint		Return				Help		Quit
Function mask	Resume	Paint		Return			View	Help		Quit
Function mask (View menu)	Resume	Paint		Return				Help		
Help menu				Return				Help		Quit
Help text	Relatd	RollFw	RollBk	Return			Menu	Help		Quit

Online Help for Keys

To obtain a list of some of the most common generic key names used in the documentation, press **KeyHlp** from the first screen that appears when you enter the MAPPER system.

Some MAPPER systems also provide a list of actual key sequences to use on your keyboard for various key actions. If this feature is available, you will see an entry in the first **KeyHlp** screen called **KeyMap**. Press the keys shown to the right of the word **KeyMap** to display this list of actual key sequences.

If **KeyMap** does not appear on the first **KeyHlp** screen, specific information for your keyboard is not available online. Refer to the printed documentation for your terminal.

You can also use the following methods to obtain the list of generic key names:

Method 1

1. Press **Help** until you reach the main help menu.
2. Tab to "Key Assignments for Documentation Key Names."
3. Transmit.

Method 2

1. After signing on to the MAPPER system, move the cursor to the control line.
2. Type **help,key**.
3. Transmit.

Use of the MAPPER Mouse

This section on the MAPPER mouse applies only to the U Series MAPPER System and the UNIX MAPPER Systems when using a personal computer as the MAPPER terminal.

PC MAPPER: *The Personal Computer MAPPER System supports a mouse; however, you configure your mouse with the native operating system on your personal computer. No other configuration is needed. See the* Planning and Installation Guide *for details.*

If your terminal is a PC running MS-DOS®, you can use a mouse to position the cursor on MAPPER screens and to select functions.

Though you can use either a two- or three-button mouse, the MAPPER mouse uses only two buttons:

- **Left button.** Moves the cursor to the place where you are pointing and transmits. This button is called **MouseMit** in the documentation.
- **Right button.** Moves the cursor to the place where you are pointing and does not transmit. This button is called **MousePos** in the documentation. You can begin typing at that point and press **Transmit** when you are ready to transmit.

Installing the Mouse on Your PC

Store the files you need for loading the MAPPER mouse software on your PC by following this procedure:

1. Obtain a diskette from your coordinator that contains these files: *mapmouse.exe* and *mapmouse.mnu.*
2. Create a directory on the C drive of your PC called *mapmouse*. At the C drive DOS prompt, enter the following:

   ```
   mkdir \mapmouse
   ```

3. Insert the diskette in drive A. At the DOS prompt, enter the following:

   ```
   copy a:mapmouse.* c:\mapmouse
   ```

MS-DOS is a registered trademark of Microsoft Corporation.

After storing the files on your PC, follow this procedure to load the MAPPER mouse software on your PC. Since MAPPER mouse software will be resident in memory, load it before loading other software to avoid memory fragmentation.

1. Install the mouse hardware and software provided by the vendor.
2. Change to the directory containing your MAPPER mouse software. At the C drive DOS prompt, enter the following:

   ```
   cd \mapmouse
   ```

3. At the DOS prompt, enter **mapmouse.**

The MAPPER mouse software loads and writes a file containing local information. Once you have MAPPER mouse software loaded in your PC, the function of the mouse buttons is unique to MAPPER software.

Guidelines

- If an error occurs when you are loading the MAPPER mouse software, a message appears. Contact your coordinator for assistance.
- If you run another application that uses your mouse, reload the MAPPER mouse before returning to the MAPPER system.
- Here are the sizes of the mouse-loading files:

  mapmouse.exe — approximately 28K bytes

  mapmouse.mnu — approximately 4K bytes

Appendix D
Control Commands for Transferring Data

You may find additional information specific to your MAPPER system by pressing **ReadMe** from the sign-on screen.

This appendix lists the data control commands available that let you control the format of the data in the files being transferred. Use these commands with the Create File (FILE) function and the START function when you do not want the files in MAPPER format.

Data Control Commands

Enter these commands, beginning in column 1, on any line in the report. They take effect from that point on but the line containing the control command is removed from the output. Except for $INCL$, $TABA$ and $TRNA$, the format is just the command itself.

Table D-1. Data Control Commands

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td>$CLRT$</td><td>Clears character and tab code translation initialized by the $TRNA$ and $TABA$ commands; translates tab characters to spaces (default).</td></tr>
<tr><td>$DATA$</td><td>Suspends any data translation by the $TRNA$ and $TABA$ commands; uses the original data (does not affect $TABA$ translations). See the following chart to determine how tab characters are affected:
<table>
<tr><td>Data control command:</td><td>Tabs after translation:</td><td>Tabs after $DATA$ command:</td></tr>
<tr><td>no command</td><td>space</td><td>tab</td></tr>
<tr><td>$TABA$ 'x'</td><td><i>x</i></td><td><i>x</i></td></tr>
<tr><td>$CLRT$</td><td>space</td><td>tab</td></tr>
</table>
(<i>x</i> = any character)</td></tr>
<tr><td>$DCML$</td><td>Deletes all asterisk type lines except heading lines.</td></tr>
<tr><td>$DFFL$</td><td>Deletes all period type lines except heading lines.</td></tr>
<tr><td>$ICML$</td><td>Includes all asterisk type lines.</td></tr>
<tr><td>$IFFL$</td><td>Includes all period type lines.</td></tr>
</table>

continued

Table D-1. Data Control Commands (cont.)

Command	Description
$INCL$	Includes reports in the runstream from numeric drawer ***nnnn***, reports ***n,n*** and ***n-n*** (any combination in any order). Format: **$INCL$ [H] D*nnnn* R{*n,n,n-n*}** or **$INCL$ [H] D*nnnn* A** where: **H** is a control character to pass report headings. ***nnnn*** is the numeric drawer designator. ***n*** is a report number. Separate individual report numbers with commas; specify a range of reports with a hyphen. **A** is a control character to include all reports in a drawer. For example, **$incl$ h d2 r3-6,23,40-57** includes reports, with headings, from numeric drawer 2, reports 3 through 6, 23, and 40 through 57.
$TABA$	Translates tab characters to the character represented by ASCII octal code ***nnn***, or to the character ***y***. Format: **$TABA$ {*nnn*|'*y*'}** where: ***nnn*** is the ASCII octal code of the character. (See Appendix B for ASCII codes.) ***y*** is the character (enclosed in apostrophes). For example, **$taba$ '&'** translates tab characters to ampersands.
1100: $TABC$ ***nn***	Translates tab characters to the character represented by Fieldata code ***nn***. Format: **$TABC$ *nn*** where: ***nn*** is the Fieldata code of the character. (See Appendix B for Fieldata codes.) For example, **$tabc$ 50** translates tab characters to asterisks.

Table D-1. Data Control Commands (cont.)

Command	Description
1100: $TRAN$	Translates character ***x*** to the character represented by Fieldata code ***nn***, or reestablishes the translation previously suspended by $DATA$. Format: **$TRAN$** ***x,nn*** or **$TRAN$.** where: ***nn*** is the Fieldata code of the character. (See Appendix B for Fieldata codes.) ***x,nn*** translates character ***x*** to the character ***nn*** in the Fieldata character set. (See Appendix B for Fieldata codes.) **.** is space-period-space (or may be blank to the end of the line) and reestablishes the translation previously suspended by the $DATA$ command. For example, **$tran$ &,05 $,03** translates ampersands (&) to spaces (Fieldata code 05) and dollar signs ($) to pound signs (#), (Fieldata code 03).
$TRNA$	Translates character ***x*** to the character represented by ASCII octal code ***nnn*** or to the character ***y*** (any combination in any order), or reestablishes the translation previously suspended by the $DATA$ command. Format: **$TRNA$** {***x,nnn***|***x,'y'***...} or **$TRNA$.** where: ***x,nnn*** translates character ***x*** to the character ***nnn*** in the ASCII character set. ***x,'y'*** translates character ***x*** to the literal ***'y'***. **.** is space-period-space (or may be blank to the end of the line) and reestablishes the translation previously suspended by the $DATA$ command. For example, **$trna$ &,040 $,'?'** translates ampersands (&) to spaces (ASCII octal code 040) and dollar signs ($) to question marks (?).

Appendix E
Mathematical Operators and Functions

You may find additional information specific to your MAPPER system by pressing **ReadMe** from the sign-on screen.

This appendix contains tables of the following:

- Mathematical operators
- Priorities assigned to mathematical operators
- Mathematical and trigonometric functions

Table E-1 lists valid mathematical operators. Values A and B can be integers, real numbers, or expressions composed of integers and real numbers.

Table E-1. Mathematical Operators

Operators	Description
+	Addition: A+B.
-	Subtraction: A-B.
/	Division: A/B.
//	Integer division: A//B. Gives the unrounded integer portion of the dividend of value A divided by value B.
*	Multiplication: A*B. Note that (A)(B) and AB are improper notations for multiplication.
**	Exponentiation: A**B. Gives the result of value A raised to the power of value B.
-	Unary minus: -A. Gives the negative of value A.

Table E-2 lists the priorities assigned to the mathematical operators. To change priorities, use parentheses. For example, 4*3+2 is processed in this order: 4*3, then 12+2. But 4*(3+2) is processed in this order: 3+2, then 4*5.

Table E-2. Mathematical Operators Priorities

Priority	Operator
First	Unary minus (-)
Second	Exponentiation (**)
Third	Multiplication (*) Division (/) Integer division (//)
Fourth	Addition (+) Subtraction (-)

Table E-3 lists and describes mathematical and trigonometric functions. Note that *x* is an integer, a real number, or an arithmetic expression.

Table E-3. Mathematical and Trigonometric Functions

Function	Description
ABS(*x*)	Absolute value or magnitude of *x*.
ACOS(*x*)	Arc cosine of *x* in radians.
ASIN(*x*)	Arc sine of *x* in radians.
ATAN(*x*)	Arc tangent of *x* in radians.
AVG(*x1,...,xn*)	Average value of all specified values or expressions, *x1* through *xn*. Use with the Calculate function only.

continued

Table E-3. Mathematical and Trigonometric Functions (cont.)

Function	Description
CBRT(*x*)	Cube root of *x*.
COS(*x*)	Cosine of *x* in radians.
CTN(*x*)	Cotangent of *x* in radians.
DEG(*x*)	*x* radians expressed in degrees.
EXP(*x*)	Natural number e raised to the power *x*.
FRAC(*x*)	Fractional portion of *x*.
HCOS(*x*)	Hyperbolic cosine of *x*.
HSIN(*x*)	Hyperbolic sine of *x*.
HTAN(*x*)	Hyperbolic tangent of *x*.
INT(*x*)	Integer portion of *x*.
LOG(*x*)	Logarithm of *x* in base e.
LOG10(*x*)	Logarithm of *x* in base 10.
MAX(*x1,...,xn*)	Maximum; largest value of all specified values or expressions, *x1* through *xn*. Use with the Calculate function only.
MIN(*x1,...,xn*)	Minimum; smallest value of all specified values or expressions, *x1* through *xn*. Use with the Calculate function only.
MOD(*x,y*)	Modulus; remainder value of *x/y*.
PI(*x*)	Value of pi (3.14159265358979). Use with the Arithmetic function only.
RAD(*x*)	*x* degrees in radians.
RAN(*x,y*)	Random integer value in range *x* to *y*. Use with the Calculate function only.
SIN(*x*)	Sine of *x* radians.
SQRT(*x*)	Square root of *x*.
SUM(*x1,...,xn*)	Total value of all specified values or expressions, *x1* through *xn*. Use with the Calculate function only.

Table E-3. Mathematical and Trigonometric Functions (cont.)

Function	Description
TAN(*x*)	Tangent of *x* radians.
VAVG(*x1*,...,*xn*)	Vertical average of all specified values or expressions, *x1* through *xn*. Use with the Calculate function only.
VMAX(*x1*,...,*xn*)	Vertical maximum of all specified values or expressions, *x1* through *xn*. Use with the Calculate function only.
VMIN(*x1*,...,*xn*)	Vertical minimum of all specified values or expressions, *x1* through *xn*. Use with the Calculate function only.
VSUM(*x1*,...,*xn*)	Vertical sum of all specified values or expressions, *x1* through *xn*. Use with the Calculate function only.

Appendix F
Application Power Tools (APT)

APT is a set of MAPPER runs that help you design, implement, and maintain your applications. For example, with these runs, you can do the following:

- Document information about applications and their components
- Share information with other users between applications
- Track the effect that database changes may have on other users' work
- Design reports and screens

APT provides powerful tools for virtually any level of MAPPER software user, including application managers, run designers, coordinators, and end users.

This section provides an overview of APT. It includes the following topics:

- Using APT
- Applications and the data dictionary
- Creating dictionary entries
- Retrieving dictionary information
- Verifying dictionary entries
- Sharing dictionary information
- Creating reports and screens

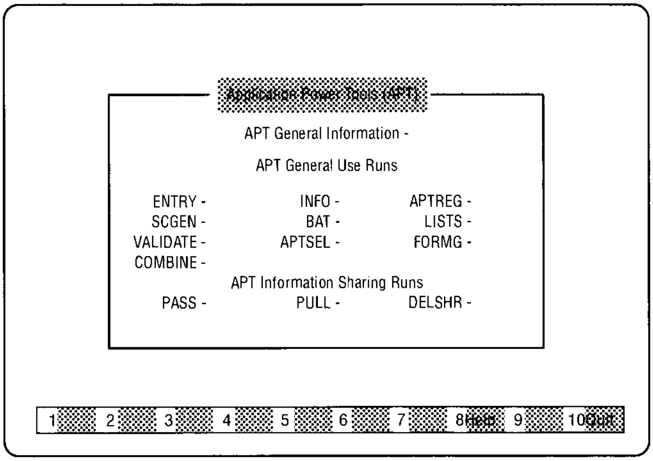
You can get more detailed information online by entering `apt` and pressing the **Help** key from any field on the menu.

Using APT

The APT runs are especially useful during implementation of an application, but you can use them during the design and maintenance phases as well.

When you create an application, the way you design, implement, and control it is up to you. You might choose to set up rigorous standards and requirements for users of your application, making full use of the APT capabilities. Or, you might find that a minimum amount of documenting and tracking best meets your needs. The extent that you want to implement APT capabilities is up to you.

You can execute the runs using the APT menu, shown below. You access this menu by entering `apt` while signed on to MAPPER software. You can obtain information about each field by pressing the **Help** key.



Each run also has a control line format; the formats are listed in Appendix A.

As you use the APT runs, you fill in forms and press function keys to complete each task. Use the **APTSEL** run to select a default application that is displayed in each form as you use the runs. For more information, press **Help** from the **APTSEL** selection.

Applications and the Data Dictionary

Before exploring the APT runs, it is important to understand the concepts of an application and the data dictionary.

What Is an Application?

A MAPPER application is the set of runs, reports, fields, and screens — known as the components of the application — that are used to accomplish a business task or operation. Each application has a unique name, which is registered by the MAPPER system coordinator. The person who defines, manages, and controls the use of the application is called the application owner.

What Is the Data Dictionary?

The APT data dictionary is a database of components of an application. Like any standard dictionary such as Webster's, the data dictionary is a tool containing definitions, usage, and other information that people can "look up." Whereas Webster's dictionary describes words in a language (including such things as usage and the parts of speech — noun, verb, and so on), the APT data dictionary describes components of a MAPPER application (describing such things as the purpose, size, location, and type of item — field, run, and so on).

When you own an application, you and the other users in your department are the authors of the dictionary. You decide what information to define and to what degree. You can define any item that can be created in MAPPER software, such as fields, reports, runs, and screens.

For example, you may enter information about fields of a given report into the data dictionary so users of your application can "look up" the fields' attributes and definitions.

Creating Dictionary Entries

To define components of your application in the data dictionary, you use the ENTRY and APTREG runs. With the ENTRY run, you define one component at a time. With the APTREG run, you define all the fields of a report or drawer.

The ENTRY Run

Use the ENTRY run to create, modify, or delete data dictionary entries. The ENTRY run is especially useful when you are designing an application and want to register the components as you create them. For example, suppose you have just created the Status Code (St_Cd) field while designing an application. As shown in Figure F-1, you can define, or *register*, details about that field in the data dictionary:

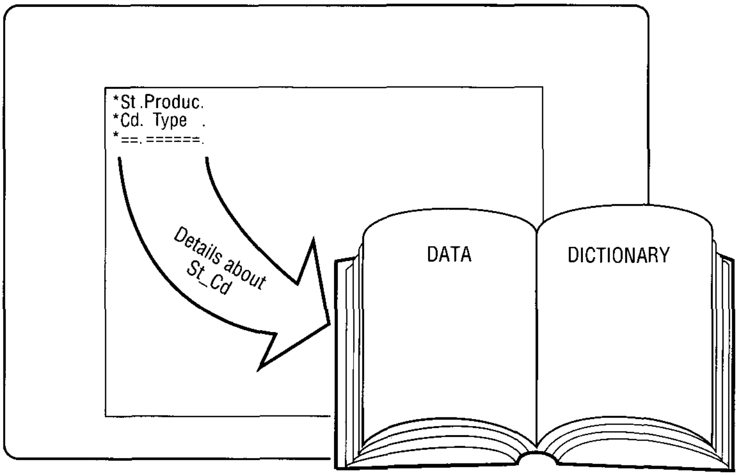


Figure F-1. Dictionary Entry for St_Cd Field

To define the component in the data dictionary, you fill in a form with information, such as the size and edit code. When the form is complete, a documentation area is displayed in which you can enter a textual description of the field. This description can include a variety of information, such as the purpose of the field or its possible values.

The APTREG Run

An efficient way to register fields from an existing application is to use the APTREG run. It provides a shortcut method for entering information about fields from an entire report or drawer. Using the APTREG run saves you time because you execute the run only once to automatically register all the fields.

If you execute APTREG against a report 0 (such as 0B), you can capture the edit codes in addition to the field headings and sizes.

After using the APTREG run, you can use the ENTRY run to enter documentation for each field. Once you have created dictionary entries, you can display them, as described in the next section.

Retrieving Dictionary Information

Use the INFO and LISTS runs to view dictionary entries. If you are a run designer, you can use the BAT (Build APT Table) run, which allows you to set up pointer tables in your runs so that you can use a powerful feature of the LISTS run.

The INFO Run

Using the INFO run, you can display a representation of the dictionary information for one or more entries. The display contains all information you entered about the entries, including the documentation.

The LISTS Run

Using the LISTS run, you can display specific types of dictionary entries; you use it like an index to the dictionary. You can display the following types of entries:

- Applications within your department
- Specific types of entries, such as all fields or all runs
- All runs and external subroutines that access a specific report
- All reports or screens that access a specific field

For example, suppose you need to extend the Status Code (St_Cd) field to three characters, but you realize that such a change to the database may impact other users' work. The LISTS run helps you identify which reports or screens are affected so that you can determine the impact of the change and, if necessary, notify the users.

The BAT Run

A powerful way to use the LISTS run is to list all runs and subroutines that access a particular report. This capability lets you determine which runs will be affected if you change or delete a report. Before you can use this capability, each run must contain an APT table, which is a list of all the cabinets, drawers, and reports accessed by the run. Run designers create these APT tables using the BAT (Build APT Table) run. For more information, see the *Run Design Reference* or press **Help** from the BAT selection on the APT menu.

Verifying Dictionary Entries

Use the VALIDATE run to verify data dictionary entries, their documentation, and the existence of APT tables; this gives you control over your application and the integrity of the data dictionary. You can check for the following:

- Entries with no documentation
- Runs, screens, and external subroutines without valid APT tables
- Runs that are not registered in the data dictionary
- Any reports registered in the data dictionary that do not physically exist on the system

As the application owner, you decide which items to check and the frequency with which to check them. For example, you might require users to enter and document every component of the application and create APT tables in all runs. On the other hand, you may find that it is only necessary to register certain items in the data dictionary. Remember that the data dictionary is only as useful as the information you and the other users put into it.

Sharing Data Dictionary Information

Your application might contain data dictionary information that could be used by another application — or several other applications. Similarly, your application might benefit from access to the data dictionaries of other applications. You can share information among applications to avoid redundant data dictionary entry and to control who is using what information.

Three runs help you control the sharing of data dictionary information: PASS, PULL, and DELSHR.

The PASS Run

You use the PASS run to share your data dictionary information with other applications on your system. You can share all the data dictionary items from your application, or you can specify which items to share. It is important to note that the PASS run shares only data dictionary information; it does not share the actual data.

For example suppose you have an application called Employees. Your data dictionary contains an entry for a field called the Social Security Number (SS_Number) field. You recently had a conversation with a coworker, who uses an application called Payroll. In your discussion, you learned that your coworker's application also uses the SS_Number field. The Payroll application owner has not created data dictionary entries for the application, but the users of Payroll could benefit from the SS_Number dictionary entry you have created. As shown in Figure F-2, you can use the PASS run to allow the Payroll application to access your dictionary information for that field:

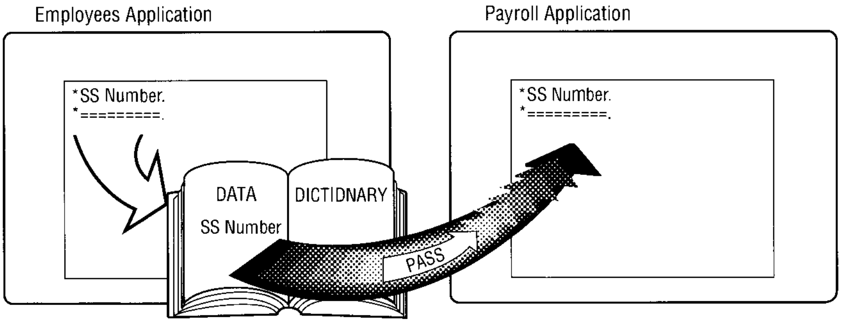


Figure F-2. Passing Dictionary Information

When you share information with another application, a link is established between your application and the receiving application. So, for example, if you used the LISTS run to find out which reports use the SS_Number field, the Payroll application would be listed.

The Public Dictionary

If any data dictionary information is needed by several applications, the owner of an application can share it with the public dictionary. The public dictionary makes the information available to any application on the system.

For example, suppose you discover that many applications, such as Payroll, Benefits, Personnel, and others, use the SS_Number field. By passing the information to the public dictionary, you make the dictionary entry available to all those applications, as shown in Figure F-3. After passing the information, you notify the owners of those applications that the information is now available to them:

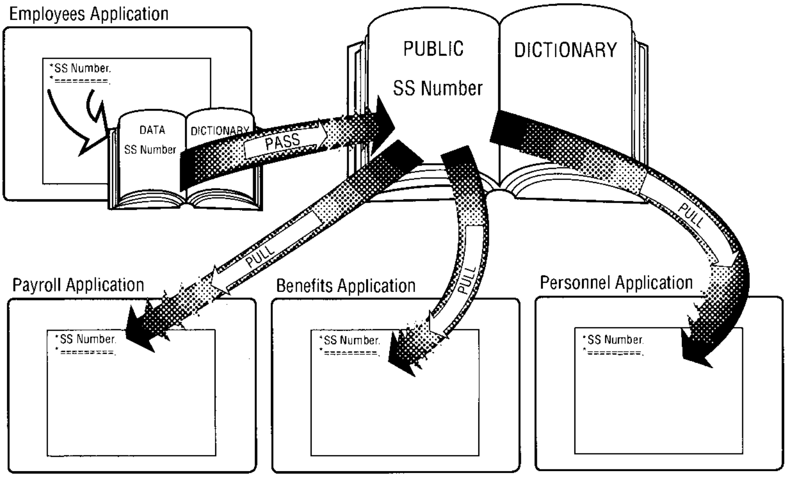


Figure F-3. Using the Public Dictionary

The PULL Run

Once you have used the PASS run to pass the entry to the public dictionary, other applications can use the PULL run to access it.

When another application pulls information you have shared with the public dictionary, the link between your application and the receiving application is established, just as if the information was passed directly.

The DELSHR Run

If other applications no longer need access to your information, you can sever the link using the DELSHR run. This makes your data dictionary information unavailable to the application it was shared with. As the application owner, you control who is using your dictionary information.

Creating Reports and Screens

Included in the APT features are the FORMG and COMBINE runs, which let you design reports to suit your needs. The SCGEN run lets you create your own menus and input screens.

The FORMG Run

The FORMG run provides a menu-driven method for creating experimental reports. It is similar to the FORMGEN run released with previous levels of MAPPER software. However, it also gives you the option of selecting items that have been registered in the data dictionary.

The FORMG run offers several advantages over the manual procedure of creating an experimental report. For example, you can use the menu-driven interface, forms, and function keys to lay out your database. You can also modify the design and rearrange fields easily.

The COMBINE Run

With the COMBINE run, you can create an ad hoc report by combining fields from two or more reports. Like the Match (MA) function, COMBINE enables you to select fields to match on and include in the receiving report. However, COMBINE includes in the report only those fields you select.

The SCGEN Run

Using the SCGEN run, you can interactively design menus and screens using menus, forms, and function keys. The SCGEN run automatically builds the code, which is accessed in a run written by a MAPPER run designer.

Interrelationships of APT Runs

Many of the APT runs have direct relationships with the data dictionary and with each other; other runs can be used independently. Figure F-4 shows the interrelationships of most of the APT runs:

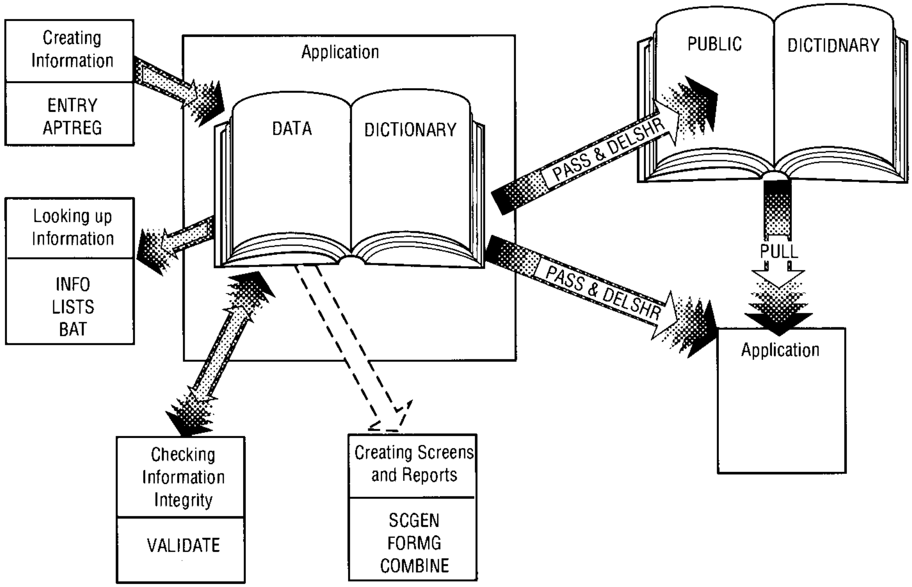


Figure F-4. Interrelationships of APT Runs

This appendix provided an overview of some of the APT runs; for more information, press **Help** from any selection on the APT menu.

Appendix G
Using Your Workstation

This section applies only to the A Series MAPPER System.

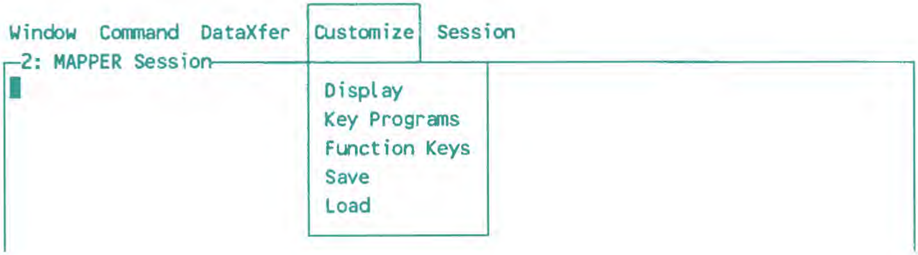
You may find additional information specific to your MAPPER system by pressing **ReadMe** from the sign-on screen.

This appendix explains how to customize your workstation to help you work most efficiently.

Customizing Your Workstation

You can modify the display of certain characters and program your keyboard, then save these configurations to use again and again.

To customize your workstation, display the menu bar and select **Customize**. The following screen is displayed:



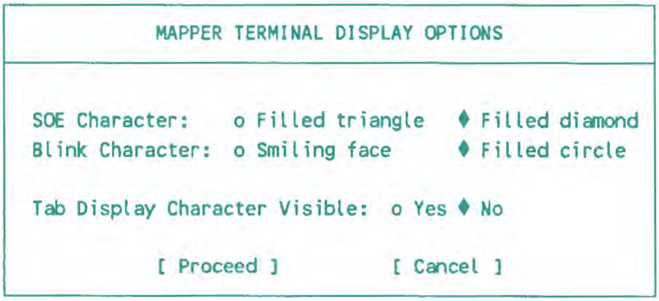
Selecting Display Options

You can modify the appearance of the SOE, blink, and tab characters by following these three steps.

Step 1

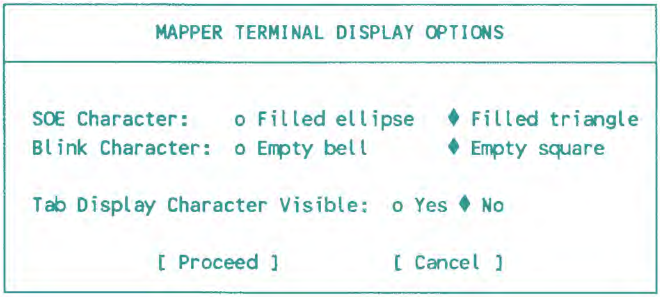
Select **Display** from the Customize menu.

On PC workstations, this form is displayed:



```
MAPPER TERMINAL DISPLAY OPTIONS

SOE Character:    o Filled triangle   ♦ Filled diamond
Blink Character:  o Smiling face      ♦ Filled circle

Tab Display Character Visible:  o Yes ♦ No

          [ Proceed ]          [ Cancel ]
```

On B20 workstations, this form is displayed:



```
                 MAPPER TERMINAL DISPLAY OPTIONS

SOE Character:    o Filled ellipse   ♦ Filled triangle
Blink Character:  o Empty bell       ♦ Empty square

Tab Display Character Visible:  o Yes ♦ No

           [ Proceed ]          [ Cancel ]
```

Step 2

Choose the display options you want. Following are descriptions of each field:

Field	Description
SOE Character	This field allows you to choose the symbol that represents the SOE character on your display. On PC workstations, you may select either a filled triangle or a filled diamond (default). On B20 workstations, you have the choice between a filled ellipse or a filled triangle (default).
Blink Character	The MAPPER system uses a blinking character to surround or highlight certain display information sent to your screen (for example, system messages). On PC workstations, the blinking character can be a smiling face or a filled circle (default). On B20 workstations, the blinking character can be an empty bell or an empty square (default).
Tab Display Character Visible	This field allows you to choose whether the tabs are visible on your screen. If you choose **Yes**, tab marks are displayed as small, mid-line dots (·). If **No**, tab marks are invisible (default).

Step 3

After choosing the display options you want, select **Proceed** to initialize the options and remove the form from your display.

If you do not want to change any display options, select **Cancel**. The display then appears as it did before you selected **Display**.

Programming Your Keyboard

You can redefine the keys on your keyboard to suit your needs. This means that, for example, you could define the keypad plus (+) key to perform the Sign On function or the minus (-) key to perform a Cabinet Switch function.

You can program all keys except those reserved for special use. You cannot program these keys:

- **Ctrl** (or **Code** on the B20)
- **Shift**
- **Action** on the B20
- All control sequences (such as C-**T**)

Note that you do not need to choose a selection from the menu bar; you can program your keys any time that you are signed on to the MAPPER system. You can make your key definitions permanent by saving them to a file; this is discussed later under "Maintaining Key Definitions." If you do not save the key definitions, they are retained only until the MAPPER Connect software is terminated or a new configuration file is loaded.

Key Programming Procedures

Use the following procedure to program your keys:

1. Press **Shift-F6**.
2. Press the key to be programmed.

3. Enter the key program (one or more keystrokes that you want associated with the key).

   Note: *If you press a key that causes a transmission to the MAPPER system, for example,* ***C-Backspace, F1, F2, F4,*** *or* ***Transmit,*** *you are finished with the key programming. When programming keys of this nature, omit step 4; do not press* ***Shift-F6*** *to finish key programming. Since key programming has already been completed, pressing* ***Shift-F6*** *would reinitialize the key programming procedure.*

4. Press **Shift-F6**.

5. Repeat steps 1 through 4 for every key that you want to redefine.

Invalidating a Key Function

If you want a key to be invalid (you do not want it to perform any function), follow these steps:

1. Press **Shift-F6**.

2. Press the key you want to have no function.

3. Press **Shift-F6**.

Restoring a Key Function

To restore the original function of a key, follow these steps:

1. Press **Shift-F6**.

2. Press two times the key you want to restore to its original function.

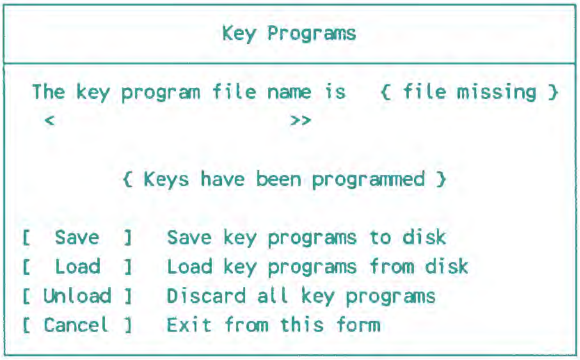
3. Press **Shift-F6**.

Maintaining Key Definitions

You can maintain your key definitions temporarily or permanently by following these two steps.

Step 1

Choose **Key Programs** from the Customize menu. The following form is displayed:



```
                  Key Programs

 The key program file name is   { file missing }
  <                      >>

        { Keys have been programmed }

[  Save  ]   Save key programs to disk
[  Load  ]   Load key programs from disk
[ Unload ]   Discard all key programs
[ Cancel ]   Exit from this form
```

Step 2

Select an option from the Key Programs form. Following are descriptions of each field:

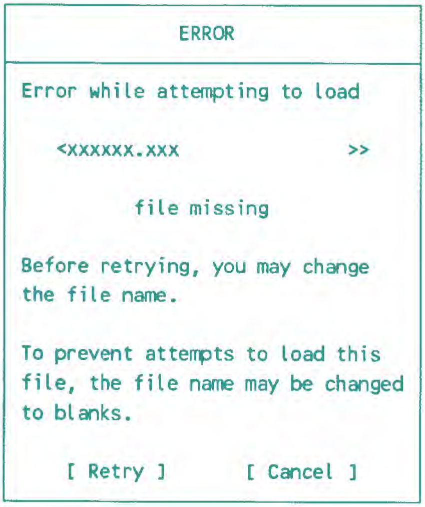
Field — **Description**

The key program file name is

This field is blank unless you have previously specified a key program file name. You may enter, change, or delete (blank out) the key program file name in this field.

file missing

This field is displayed if you attempt to load a nonexistent key program file name. You receive a warning (three short beeps), **file missing** is displayed in this field, and the following Error form is displayed (in the following example, xxxxxx.xxx represents a nonexistent file name):



```
                 ERROR

Error while attempting to load

   <XXXXXX.XXX            >>

          file missing

Before retrying, you may change
the file name.

To prevent attempts to load this
file, the file name may be changed
to blanks.

    [ Retry ]        [ Cancel ]
```

When the Error form is displayed, you have two choices. You can change the file name and select **Retry**. Or you can select **Cancel**, which returns you to the Key Programs form where the file name you attempted to load is shown in **The key program file name is** field. At this point you can specify another file or blank out the field.

continued

continued

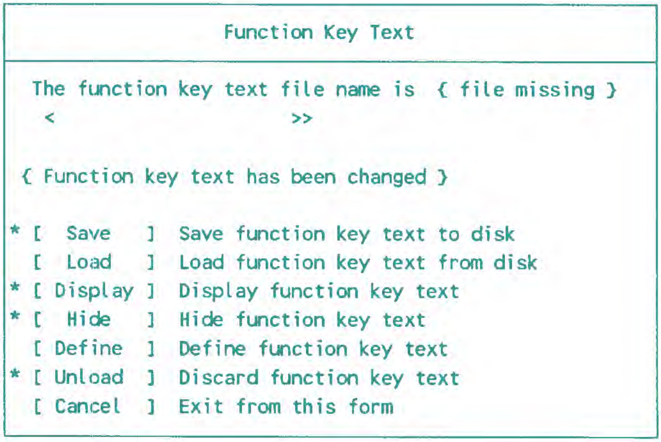
Field	Description
Keys have been programmed	This field is displayed only if you have modified the key definitions and have not saved them to a file.
Save	Select this field to save key definitions to a file.
Load	Select this field to load key programs from a previously saved file. ***Note:*** *Selecting* **Load** *prior to selecting* **Save** *causes all keys to return to their original functions before new key values were assigned.*
Unload	Select this field to cause the keyboard keys to return to their default (normal, unprogrammed) state. ***Note:*** *It is not necessary to unload the current key programs before loading a new set of key programs (refer to the* ***Load*** *description).*
Cancel	Choose this field before selecting **Save**, **Load**, or **Unload** to remove the Key Programs form without any subsequent operation taking place. It returns you to the MAPPER Session window.

Defining Function Key Text

You can specify a word associated with each function key to be displayed in a legend at the bottom of your screen. Follow these steps.

Step 1

Select **Function Keys** from the Customize menu. The following form is displayed:



```
                       Function Key Text

  The function key text file name is  { file missing }
   <                     >>

 { Function key text has been changed }

* [  Save    ]  Save function key text to disk
  [  Load    ]  Load function key text from disk
* [ Display  ]  Display function key text
* [  Hide    ]  Hide function key text
  [ Define   ]  Define function key text
* [ Unload   ]  Discard function key text
  [ Cancel   ]  Exit from this form
```

* **Save**, **Display**, **Hide**, and **Unload** appear only if function key text exists; otherwise, these fields are not visible.

Step 2

Select an option from the Function Key Text form. Following are descriptions of each field:

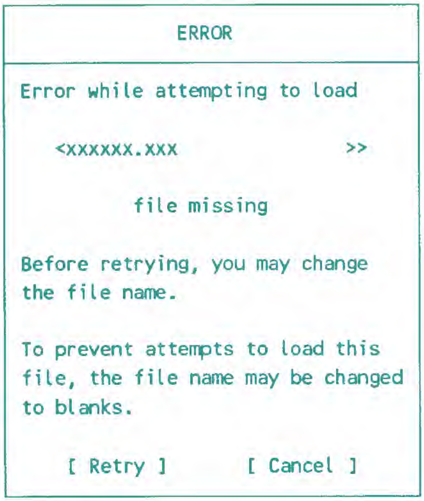
Field | **Description**

The function key text file name is

This field is blank unless you have previously specified a function key text file name. You may enter, change, or delete (blank out) the file name in this field.

file missing

This field is displayed if you attempt to load a nonexistent function key text file name. You receive a warning (three short beeps), **file missing** is displayed in this field, and the following Error form is displayed (in the following example, xxxxxx.xxx represents a nonexistent file name):



```
                ERROR

Error while attempting to load

   <xxxxxx.xxx              >>

          file missing

Before retrying, you may change
the file name.

To prevent attempts to load this
file, the file name may be changed
to blanks.

    [ Retry ]       [ Cancel ]
```

When the Error form is displayed, you have two choices. You can change the file name (the default name is MAPPER.FKY) and select **Retry**. Or you can select **Cancel**, which returns you to the Function Key Text form where the file name you attempted to load is shown in **The function key text file name is** field. At this point you can specify another file, use the default name, or blank out the field.

continued

continued

Field	Description
Function key text has been changed	This field is displayed only if the current text was actually changed and has not yet been saved.
Save	Select this field to save the currently defined function key text to the file you have specified. The form is removed from the display and you are returned to the MAPPER Session window.
Load	Select this field to load the existing function key text for the present workstation configuration. The function key text automatically is displayed at the bottom of your display, and the form remains visible. It is not necessary to choose Display from the Customize menu to cause the legend to appear.
Display or **Hide**	If you have specified a valid function key text file name, either **Display** or **Hide** is displayed, whichever is appropriate. **Display** causes the currently defined function key text to become visible on your display. **Hide** causes the currently defined function key text to become invisible or hidden from view. The function key legend is removed from the display only; it is not totally discarded. (If you wish to discard the currently defined function key text, select **Unload**.)

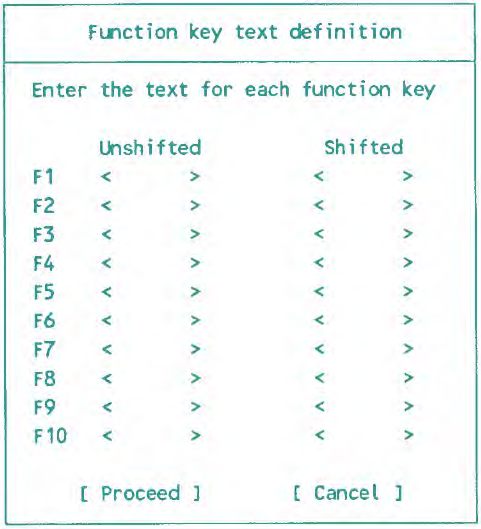
continued

continued

Field **Description**

Define

Select this field to define the function key text (up to seven characters for each key) that is displayed on the bottom of your screen. The following form is displayed:



```
          Function key text definition

   Enter the text for each function key

          Unshifted           Shifted
   F1     <       >           <       >
   F2     <       >           <       >
   F3     <       >           <       >
   F4     <       >           <       >
   F5     <       >           <       >
   F6     <       >           <       >
   F7     <       >           <       >
   F8     <       >           <       >
   F9     <       >           <       >
   F10    <       >           <       >

        [ Proceed ]         [ Cancel ]
```

If there is any function key text in memory, the current values are displayed in the appropriate fields. To change the definitions, move to the appropriate function key field and enter the desired text. If no values are displayed, enter the text in the appropriate fields. You can enter both shifted and unshifted characters.

Select **Proceed** to return to the Function Key Text form. Any text defined in the Function key text definition form is displayed in the legend appearing at the bottom of your screen. Selecting **Cancel** any time prior to selecting **Proceed** discards the Function key text definition form and returns you to the Function Key Text form. The display then appears as it did prior to selecting **Define**.

continued

continued

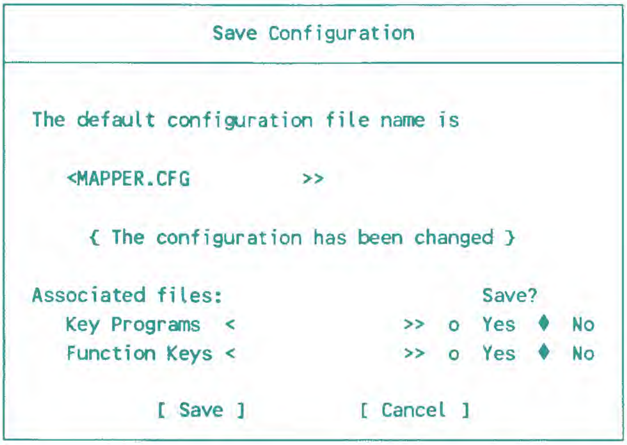
Field	Description
Unload	Select this field to erase the currently defined function key text. For the keys to be redisplayed, you must choose **Load**. (If you wish to remove the function key text from the display only, without totally discarding it, select **Hide**.)
Cancel	Select **Cancel** before selecting **Display**, **Hide**, **Define**, or **Unload** to remove the Function Key Text form without any subsequent operation taking place. You are then returned to the MAPPER Session window.

Saving Your Workstation Configuration

To save the current configuration of your workstation (that is, the set of display options and key programming files), follow these steps.

Step 1

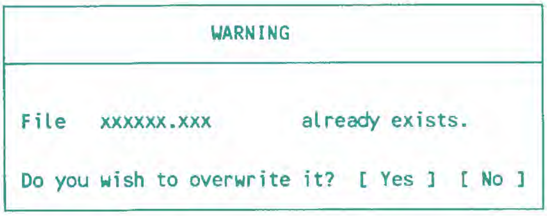
Select **Save** from the Customize menu. The following form is displayed:



```
                  Save Configuration

 The default configuration file name is

    <MAPPER.CFG         >>

      { The configuration has been changed }

 Associated files:                       Save?
    Key Programs  <              >>  ○  Yes  ◆  No
    Function Keys <              >>  ○  Yes  ◆  No

            [ Save ]           [ Cancel ]
```

Step 2

Fill in the fields you want, then select **Save**. Following are descriptions of each field:

Field	Description
The Configuration file name field is	This field allows you to modify the current configuration file name. This file name is used for retrieving or storing the current workstation configuration.
The configuration has been changed	This field is displayed only if you have modified the workstation configuration and have not saved it to a file.
Associated files: **Key Programs** **Function Keys**	This field allows you to specify which key programs and function key text files are associated with this configuration of the MAPPER Terminal. Although these files are stored separately from the MAPPER configuration file, they can be specified here and are loaded automatically when this configuration is loaded. (See "Maintaining Key Definitions" and "Defining Function Key Text" in this section.)
Save	Select **Save** to save the current configuration file as specified in this form, along with any associated key programs and function key text. If you change the configuration of a file and attempt to save it under a file name that already exists, the following warning screen is displayed:  WARNING File xxxxxx.xxx already exists. Do you wish to overwrite it? [Yes] [No] **Yes** is the default selection. If you select **No**, you are returned to the Save Configuration form, where you can enter another name.
Cancel	Select **Cancel** to discard the Save Configuration form and return the display to its previous state.

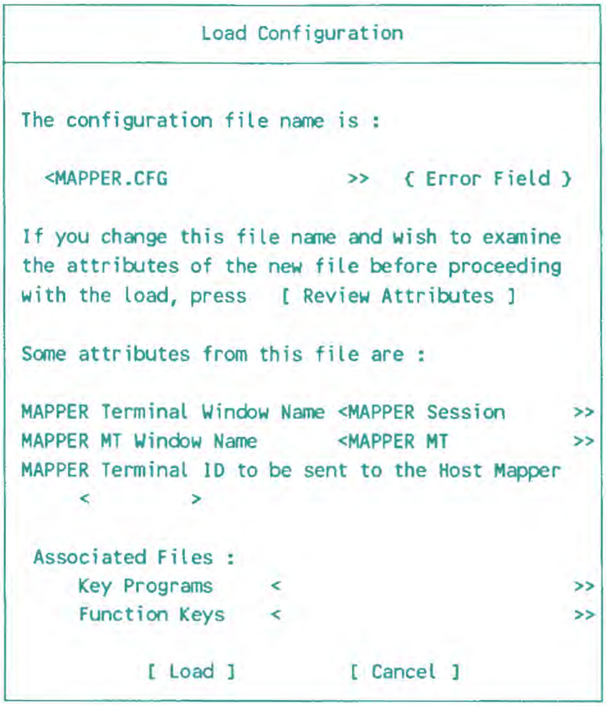
Note: *You can make temporary changes to the terminal configuration; it is not necessary to save the configuration information in order for it to take effect. Configuration changes go into effect as soon as the configuration form is processed and remain in effect until you change them, load a different configuration, or discard the MAPPER Connect software.*

Loading Your Workstation Configuration

To load a workstation configuration, follow these steps.

Step 1

Select **Load** from the Customize menu. The Load Configuration form is displayed. Following is the default Load Configuration form:



```
                    Load Configuration

The configuration file name is :

  <MAPPER.CFG               >>   { Error Field }

If you change this file name and wish to examine
the attributes of the new file before proceeding
with the load, press   [ Review Attributes ]

Some attributes from this file are :

MAPPER Terminal Window Name <MAPPER Session      >>
MAPPER MT Window Name       <MAPPER MT           >>
MAPPER Terminal ID to be sent to the Host Mapper
     <         >

 Associated Files :
     Key Programs     <                          >>
     Function Keys    <                          >>

          [ Load ]          [ Cancel ]
```

Step 2

Update the configuration file name (if desired).

Step 3

Select **Load**. The workstation configuration is now loaded and ready to use.

Glossary

A

A Series data file

A file formatted in such a way that it can be read by an A Series editor, such as Command and Edit language (CANDE).

abort

To terminate execution of a MAPPER function or run.

accesscode

An identification code subordinate to a usercode. An accesscode is used to further establish a user's identity, control security, and restrict access to disk files. *See also* log on.

active screen

The screen that shows you have signed on. The active screen looks like the sign-on screen, but your user-id takes the place of the word sign-on and the bottom line indicates which cabinet you are accessing. *Contrast with* sign-on screen. *See also* caret, sign on.

American Standard Code for Information Interchange (ASCII)

A set of numeric codes that define a character set. The ASCII character set used in MAPPER software is called full character set (FCS). *Constrast with* Fieldata, limited character set. *See also* character set, full character set.

Application Power Tools (APT)

A set of MAPPER utility runs that help you design, implement, and maintain your applications. With these runs, you can document information about applications, share information with users of other applications, and design reports and screens. *See also* data dictionary.

arithmetic expression
A single numeric value or a combination of two or more values and one or more arithmetic operators. *See also* arithmetic operator.

arithmetic operator
A special character (such as +, -, *, and =) that specifies a mathematical relationship between two values. *See also* arithmetic expression, expression.

ASCII
See American Standard Code for Information Interchange.

asterisk line
A line beginning with an asterisk (*) in column 1; it can be used as a comment line. It is not controlled by the tab positions and input edit codes of report 0. It has these characteristics: it can be shifted, it can extend to 256 characters, and it can be displayed and processed in different formats. *See also* line type.

auxiliary device
A peripheral device connected to a terminal, such as a printer. *See also* peripheral device.

B

B20 operating system (BTOS)
The operating system used when running INFOVIEW II applications on Unisys B20 series systems. *See also* INFOVIEW II.

background run
A MAPPER run that frees your terminal for other uses while it executes.

basic format
The unshifted columns of a report as defined in report 0 from left to right. On 80-character screens, the basic format contains the leftmost 80 characters of a report; on 132-character screens, the basic format includes the leftmost 132 characters of a report. Format = 0.

binary find process

A method of finding an item quickly in a sorted list. Instead of scanning data line by line, a binary find process samples the data at midpoint and continues dividing and sampling the data until it finds the target item.

binary synchronous communications (BSC)

A communications protocol for sending data to an IBM host computer.

Boolean logic

A system of logical comparisons named after mathematician George Boole. Boolean logic uses relational operators to evaluate expressions as true or false. *See also* conditional statement, IF/THEN/ELSE statement, relational expression, relational operator.

BSC

See binary synchronous communications.

BTOS

See B20 operating system.

BTOS II

The Unisys workstation operating system for the B2x and B3x series workstations.

built-in function

A predefined mathematical or other kind of operation, such as an SQRT(a) operation that calculates the square root.

C

cabinet

A group of eight drawers (B through I), referred to by number and usually used by a department. Each user signs on into a specific cabinet, assigned by the MAPPER system coordinator. Users in a particular department may have one or more cabinets to work in. *See also* department, drawer.

cabinet owner
The user responsible for defining the group of users who are permitted access to a specific cabinet. *See also* cabinet.

cabinet pair
A set of two cabinets, such as 0/1. Each cabinet in the pair holds exactly the same data, but the even-numbered cabinet allows reading and updating of reports, while the odd-numbered cabinet allows read access only. *See also* cabinet.

cabinet password
See password.

calculator
A feature of MAPPER software used by the Arithmetic (A) function that allows you to perform calculations. It can be used with a report containing predefined equations or independently, similar to a pocket calculator.

call
The abbreviation you use to specify a function, run, or run statement. For example, CAL is the call for the Calculate function.

CANDE
See Command and Edit language.

caret
A special character (^) that releases a displayed report or message, then displays the MAPPER system active screen. *See also* active screen.

case sensitivity
Pertaining to the differentiation between uppercase and lowercase letters. When describing UNIX file names, the term case sensitive means that an uppercase letter is treated as a different character than its lowercase letter.

character hierarchy

The relationship of characters to one another. For example, in limited character set, numeric characters have higher values than alphabetic characters. In sort or search-in-range processes, alphabetic characters come before numeric characters.

character set

The characters allowed in the reports in a drawer. *See also* American Standard Code for Information Interchange.

character string

A series or group of connected characters.

column

A character position in a report; for example, the first character position on the left side of a report is column 1.

column sensitive

Pertaining to specific column locations in a report.

column shifting

Moving the screen images of a report to the right or left temporarily by using the Shift function.

column-formatted report

A report having a layout identical to other reports in the same drawer. The fields are separated by tab characters. *Contrast with* freeform report.

command

An instruction entered into a report to carry out operations with a function, such as GOC commands processed by the Generate Organization Chart (GOC) function, or data control commands used with the Create File (FILE) function or Create File (FIL) run statement.

Command and Edit language (CANDE)

A message control system (MCS) that prepares and updates files in an interactive, terminal-oriented environment. *See also* message control system.

Communications Management System (COMS)
A general message control system (MCS) that supports a network of users and provides them with a consistent online interface to the host system. *See also* message control system, online.

communications output printer (COP)
Any type of printing device connected to a terminal.

COMS
See Communications Management System.

conditional statement
A statement of the Calculate (CAL) function that tests expressions and controls processing of individual equations, depending on whether a specified condition is true or false. The four conditional statements are IF:, THEN:, ELSE:, and FIRST:. *See also* Boolean logic, IF/THEN/ELSE statement.

configuration
(1) The particular set of connections between devices, such as terminals and printers, that allow them to function as a unit. (2) The selection of parameters that determines how software operates.

console
A display terminal used to execute functions for an operating system.

constant label
A value label that has a predefined value, such as PI. It is used in mathematical functions such as the Calculate (CAL) function. *See also* field label, value label.

control characters

A set of characters that control the format of text in word processing reports. Control characters consist of a tilde (~) and another specified character. *See also* tilde.

control line

The top line of the screen containing control positions. *See also* control position.

control line procedure

The method of executing a function in which you type the function format on the control line and transmit. *Constrast with* menu procedure. *See also* control line.

control position

The position on the top line of the screen following the SOE character (♦) after the Line and Roll fields. *See also* control line, start-of-entry character.

control sequence

A keystroke that includes pressing the **Ctrl** key. For example, the keystroke **Ctrl-I** means to hold down the **Ctrl** key while pressing the I key.

coordinator

See MAPPER system coordinator.

coordinator's department

The department in which the MAPPER system coordinator and users with coordinator privileges are registered.

COP

See communications output printer.

cumulation

The process of adding a quantity to a total and saving the new total in another field, repeated a number of times. For example, the Totalize (TOT) function can be used to cumulate a field. *See also* subcumulation.

cursor

The character (▮) on the screen that can be moved anywhere to show where to enter data. The cursor shows your current location on the screen.

cut

To mark the data to move, paste, or delete. *See also* move, paste.

cut control

A term used to indicate that your report is under the control of the CUT function. The words Cut Rid are displayed in the rightmost field of the control line.

D

data dictionary

A tool used with the Application Power Tools (APT) features that contains information about MAPPER applications. For example, the data dictionary can contain information about fields of a given report, including a textual description of each field, its size, location, and edit code.
See also Application Power Tools.

data line

Any report line below the heading divider line. *See also* line type.

data name

The name of a cabinet, drawer, or report that has been defined by using the NAME run. This name is stored in the system directory. *See also* system directory.

data unit

A unit of related information consisting of a tab line and the following lines up to the next tab line. A data unit can start with other line types when using the Search function or run statement with the U option. *See also* paragraph.

database
The cabinets, drawers, and reports maintained in files by the MAPPER system.

date format
A format that defines how a date is to be displayed. For example, DD MMM YY displays a date as 14 JUN 90.

date input specification
The code to specify the format of a date to process.

date line
The first line (line 1) of each report, showing the date and time of the last update, report number, report creation date, and user-id of the last person to update the report. The date line is a period line and is counted as one of the heading lines. *Synonym for* line 1.

date output specification
The code used to represent the format of a result date.

decode
To translate an encoded report into a readable report, using the Decode Report (DECODE) function. *See also* encode.

default
A preset value or condition that the system uses whenever you do not choose a specific selection.

default MAPPER site
The MAPPER site installed first on a host computer. *See also* site.

delimiter
A character placed on each side of a character string to indicate the beginning and ending of the character string. The delimiter can be any character other than those in the character string.

demonstration database
The reports, usually in cabinet 0, in which you can practice manual functions and run statements.

department
A group of MAPPER system users specified by a number. You specify your department number when you sign on to the MAPPER system. *See also* user registration.

device
See peripheral device.

disk pack
A disk that consists of multiple platters stacked vertically on a central spindle. Data on a disk pack are accessed by movable read/write heads. Some disk packs are removable. *Synonym for* pack. *See also* pack family.

display
(1) The terminal screen you look at while using MAPPER software. (2) To present data on the screen.

documentation report
A report in a drawer, usually report 1, that documents information about that drawer, such as the field definitions of the headings. The person who designs the new drawer should create the documentation report.

double function mask
A screen with the issuing report headings on the upper mask and the receiving report headings on the lower mask. *See also* function mask, issuing report, receiving report.

downline load
To copy a file or other information from a computer system to your own terminal.

drawer

A group of reports in a cabinet. All reports within a drawer have the same headings and line length. Each drawer of a cabinet is identified by a letter from B to I. Drawer A is accessible to all cabinets in a MAPPER system. *See also* cabinet, freeform drawer, report 0.

drawer definition

The process of designing a drawer: defining fields, data formats, edit code descriptions, report organization, display formats, and any special use rules.

drawer letter

A letter (A through I) used to identify a drawer in a cabinet. *See also* cabinet, drawer.

drawer number

The octal number that identifies the drawer and cabinet within the system. Each drawer has a unique number, such as 0010 for drawer E in cabinet 0. *See also* cabinet, drawer.

drawer password

See password.

drawer table of contents

The screen produced by the Drawer Table of Contents (T) function that lists what is contained in drawers B through I of the current cabinet.

E

ECL

See Executive Control Language.

edit code

See input edit code.

edit function

A function, such as SOE Update and Add Line, used to alter report lines.

empty report
A report that contains headings but no data.

encode
To transform a report into code using the Encode Report (ENCODE) function, making it unreadable unless the correct key is specified. *See also* decode.

end report line
The last line of a displayed report or result.

equation
Two arithmetic expressions separated by an equal sign (=).

equation operator
See operator.

equation set
The function mask and equations used with the Iterative Calculate (ICAL) run. *See also* function mask.

error message
See system message.

Exec
See Executive system.

execute
To process a particular instruction or task. Manual functions, runs, and run statements can be executed.

Executive Control Language (ECL)
The computer language used to communicate with the Executive system (Exec). *See also* Executive system.

Executive system (Exec)

A program that controls the execution of other routines. As the operating system, the Exec is the principal interface between the user and the system as a whole. *See also* Executive Control Language.

experimental report

A freeform report used to generate a new drawer. It contains the title, headings, input edit codes, formats, and predefined lines that will reside in report 0 for the new drawer.

exponential notation

The use of symbols to indicate the number of places to move the decimal. For example, 12000 in exponential notation is 12E3.

expression

A series of one or more field labels, values, or operators that produces a single arithmetic or logical value. For example, in the Calculate (CAL) function, an expression is the sequence of field labels and arithmetic operators (+,-,*, or /) that completes an equation. *See also* arithmetic expression, relational expression, relational operator.

F

FCS

See full character set.

FCSU

See full character set upper.

field

(1) A series of one or more columns of a report that are defined as an entity, such as a status code or a shipping date. (2) A selection from a menu or a position in a system message where data is entered (for example, the **Report** field in a function form). *See also* function form, menu.

field control characters (FCC)

The characters that control the type of data that the user enters in fields of the screen presented by the run.

field headings

The column headings for report fields. These titles appear above each field at the beginning of a report, immediately preceding the heading divider line. *Synonym for* headings. *See also* heading divider line.

field label

A single alphabetic character that identifies a single field or several fields for equations in mathematical functions. *See also* constant label, value label.

field name

A name that identifies a field of a report. The name is derived either from the two heading lines preceding the heading divider line of the report or from report 0 if the entire drawer is being processed. *See also* heading divider line, report 0.

Fieldata

A set of codes that define a character set. The Fieldata character set used in MAPPER software is called limited character set (LCS). *Contrast with* American Standard Code for Information Interchange, full character set. *See also* character set, limited character set.

floating-point number

A fractional number, either positive or negative, that contains a decimal point.

format

(1) One of several variations of a report within a drawer (for example, basic format, format 1, format 2, and so on), each of which displays a different selection of columns of data in the report. The formats are defined in the report 0 of that drawer. (2) The specific fields used in manual functions, which vary from one function to another. *See also* report 0.

format lines

The lines in report 0 of a drawer that specify the report columns displayed in different report formats. *See also* drawer, format, report 0.

freeform drawer

A drawer used for freeform reports. Drawer A is a freeform drawer accessible from all cabinets in the system. *See also* drawer, freeform report.

freeform report

A report without a columnar structure determined by the report 0 of its drawer. You can use freeform reports for memos, bulletins, run control reports, or informal columnar reports. *Contrast with* column-formatted report. *See also* report 0.

full character set (FCS)

The character set that allows uppercase and lowercase letters, stored internally as ASCII characters. *See also* American Standard Code for Information Interchange, character set.

full character set upper (FCSU)

The set of ASCII codes in uppercase only. *See also* American Standard Code for Information Interchange, character set, full character set.

function

An operation you perform on one or more reports or on a result. Examples of functions are Match (MA), Search (S), or Sort.

function bar

An information bar at the bottom of the MAPPER active screen. The ten key names correspond to the **F1** to **F10** function keys. *See also* function keys.

function call

See manual function call.

function form

A screen, resembling a menu, that contains fields in which you supply information needed to perform a manual function. *See also* menu.

function keys

The set of keys on your terminal keyboard (for example, **F1** or **F2**) programmed to perform operations when you press them. *See also* function bar.

function mask

A screen of field headings below which you enter parameters as instructions for a manual function. *See also* double function mask, parameter.

G

granularity

A measurement used when allocating storage space to a mass storage file. For example, the Element (ELT) function assigns a file with a maximum granularity of 262143 tracks. *See also* mass storage, mass storage file.

H

heading divider line

The line beginning with an asterisk and made up of equal signs and periods that separates the field headings from the data; counted as one of the heading lines. Many functions rely on the heading divider line to determine where field headings end and data begins. *See also* field headings, field name, heading lines.

heading lines

The lines including and following the date line that show the drawer letter and report number, the names of fields, and the heading divider line. *Synonym for* report headings.

headings

Synonymous with field headings.

HELP run

A MAPPER run that supplies online information about functions, run statements, system messages, and other MAPPER operations.
See also online.

home position

The upper left corner of the screen.

horizontal operation

A computation performed in one or more fields across each data line in a report. *See also* vertical operation.

host computer

The computer that holds the database and software of a particular MAPPER site. *See also* site configuration.

I

IF/THEN/ELSE statement

A conditional statement that controls further processing based on the value of the expression in the IF: statement. If that expression is true, the expression in the THEN: statement is processed. If the expression is false, then the ELSE: expression is processed. *See also* Boolean logic, conditional statement.

INFOVIEW II

A sophisticated and flexible software tool that allows a workstation to interface to a host computer. Your PC or B20 workstation uses INFOVIEW II to access the A Series MAPPER system. *See also* B20 operating system.

input edit code

A numeric code (0 through 9) in report 0 of each drawer specifying what kind of data is allowed in each character position of the report. The person designing the drawer enters the input edit codes on the input edit line of the experimental report. *See also* input edit line, report 0.

input edit line

Line 32 in report 0 of a drawer that is composed of input edit codes and blanks, specifying what kind of data is permitted in each character position of the report. *See also* input edit code, report 0.

integer

A whole number with no fraction or decimal part. For example, the numbers 1 and 2.

issuing report

The report from which data is taken when using a manual function that processes two reports, such as the Add On (ADON) or Match (MCH) functions. The issuing report is the upper mask of a double function mask. *Contrast with* receiving report. *See also* double function mask.

J

job

A group of one or more tasks, usually processed from and under the control of a single program. A job is assigned a number by the system and treated as a discrete unit of work by the computer.

justify

To position data within a field, variable, or report. For example, if data in a field is left-justified, it begins in the leftmost column of that field.

K

Kanji characters

A character set used by several Asian languages, including Japanese and Chinese.

L

label

A name for a data value in an arithmetic expression.

label table definition lines

The lines at the beginning of a run that indicate the location of each label in the run. *See also* label.

LCS

See limited character set.

limited character set (LCS)

The character set that allows uppercase letters only, stored internally as Fieldata characters. *Contrast with* American Standard Code for Information Interchange, full character set. *See also* character set, Fieldata.

line 0

The first line of a report, never visible on the screen, containing system information about the report: the drawer letter, report number, write password, read password, number of heading lines, number of lines of the report, and depending on the system being used, the language of the report. This information is stored with the report and can be displayed with the Line Zero (LZ) function.

line 1

Synonymous with date line.

line modification function

See edit function.

line type

A type of data line in the MAPPER database, specified by a line type designator in column 1. *See also* data line, line type designator.

line type designator

A character in column 1 of a report line to specify the line type. There are four designators, indicating four line types: tab (□) — column-formatted, edited line (□ represents a tab character); asterisk (*) — column-formatted, nonedited line; period (.) — comment, nonedited line; and special (any valid character) — column-formatted edited line (may not start with a tab character, asterisk, or period). *See also* line type.

literal representation

The explicit and actual value of an item; information is interpreted exactly as it appears. For example, the literal representation of variable V11 is the characters V11, not the value that the variable holds.

log on

To access the A Series system by entering information that identifies you to the system. This data includes a valid usercode, password, and accesscode. Note that this is not the same as sign on, which is a procedure used to access the MAPPER system. *See also* accesscode, usercode.

M

manual function

A MAPPER software command, such as Search (S) or Totalize (TOT).

manual function call

The abbreviation used to request a function (for example, LOC for the Locate function).

MAPPER calculator
See calculator.

MAPPER Connect software
An INFOVIEW II application that emulates terminal functions necessary to access A Series MAPPER software. *See also* INFOVIEW II.

MAPPER format
The arrangement of data in a native system file or element that has a fixed line length and contains a transparent line 0, line numbers, and end-of-line control characters. The last line is the visible end report line.

MAPPER Session window
The INFOVIEW II window through which you access MAPPER software. *See also* INFOVIEW II.

MAPPER site
See site.

MAPPER software
The multiactivity real-time program that creates an end-user environment for file management and report generation. *See also* MAPPER system.

MAPPER system
A file management system that allows the user to maintain and manipulate a large amount of data in a report-structured database.

MAPPER system coordinator
The person who manages the database of the MAPPER system. This person configures the system, registers new users and new or updated runs, and coordinates the system user group.

MAPPER system logo
See active screen, sign-on screen.

MARC
See Menu-Assisted Resource Control.

mask line
The line of asterisks just below the heading lines in a function mask. It divides the fields for use by the function. *See also* function mask, heading lines.

mass storage
Peripheral devices that can store large amounts of data. *See also* granularity, mass storage file, peripheral device.

mass storage file
A file used as a secondary storage device that provides current report information on a specific program segment. *See also* granularity, mass storage.

menu
A list of items on a screen from which one item can be selected, either by tabbing to the selection and transmitting, or by typing in a letter, number, or character string, and then transmitting.

menu path
The sequence of menu selections you choose to request a function. The menu path usually begins with the Select Task menu.

menu procedure
The method of executing a function in which you select the function from a menu. *Contrast with* control line procedure. *See also* menu.

Menu-Assisted Resource Control (MARC)
A menu-driven interface and transaction processor for users and operators of Unisys A Series systems.

message control system (MCS)
A program that controls the flow of messages between workstations, application programs, and the operating system. *See also* Command and Edit language, Communications Management System.

message log
A record of the messages sent but not yet delivered.

message wait signal
A signal indicating that your workstation has a incoming message. When a message wait signal occurs, Msg is displayed at the top of the screen and the terminal beeper sounds. You press the **Message** key or use the Message Waiting (MSG) function to display the incoming message.

move
To remove a block of data within a report or result and place it in a new location; the data disappears from its original location. *See also* cut, paste.

multiple-field label
A label that identifies more than one report field when using mathematical functions. *Contrast with* single-field label.

N

named report
A report that has been named using the NAME run. For example, report 6B0 could have a name of Inventory.

NEWUSER database
See demonstration database.

nontab line
Synonomous with special line.

O

offline storage

A mass storage device used to hold backup data from your MAPPER system.

online

Pertaining to the availability of a process or operation while another is in progress. For example, while using the Sort function, you may access online HELP. *See also* HELP run.

operating system (OS)

A collection of computer programs that control the operation of a computer and peripherals. *See also* Executive system, peripheral device.

operator

A symbol that specifies a calculation or comparison to be performed in a function such as Calculate (CAL). *See also* arithmetic operator, relational operator.

option

A selection that you can make for special operation of a manual function. If no option is used, the default operation occurs.

option field

The leftmost position on the second screen line of a function mask where characters are entered to indicate selected variations in manual functions. *See also* function mask.

OS 1100

See Executive system, operating system.

P

pack

See disk pack, pack family.

pack family

A disk or a collection of disk packs on which physical files are stored. It is given a name of up to 17 alphanumeric characters assigned during installation and is generally designated for specific applications and users. *See also* disk pack.

paint

To refresh the screen as it was immediately after the last SOE update by pressing **Paint**.

paragraph

A group of data lines starting at either the type of line you are processing or the type of line specified with the Search U option. This includes subsequent lines up to, but not including, the next occurrence of the type of line processed or specified. *See also* data unit, line type.

parameter

An item of information supplied to a function or run to indicate specific values or fields to process. *See also* function mask.

password

A character string used as a security feature to prevent unauthorized access to information on the MAPPER system. There are five kinds of passwords: the sign-on password restricts access to the MAPPER system; the write password restricts updates to reports; the read password restricts users from reading a report; the drawer password restricts access to specific drawers within a cabinet; and the cabinet password restricts access to a specific cabinet. *See also* user registration.

paste

To insert cut data within the same report or result or into another report or result. *See also* cut, move.

path name

The complete UNIX file identifier, including all directories, subdirectories, and the file name.

period line

A line beginning with a period in column 1; it can be used as a comment line. It is not controlled by the tab positions and input edit codes of report 0. It cannot be shifted, it is limited to the screen display size, and it cannot be displayed or processed in different formats. *See also* line type.

peripheral device

Any data communications or input device attached to a terminal or host computer. For example, an AUX printer is a peripheral device attached to a terminal; a modem is a peripheral device attached to the host computer. *See also* mass storage.

predefined equation

A formula stored in a report that can be displayed so you can fill in the values without retyping the equation. It is used with the Arithmetic (A) function.

predefined lines

The lines in report 0 that contain tab positions, preset data, or reserved words. Each report 0 can have multiple predefined lines for which you can specify data or values that automatically appear when a new line is added to a report. These are data lines that have certain fields already filled in for permanent unchanged data and other fields left blank for entering changeable data. *See also* data line, report 0.

purge

The process of copying an entire database to tape. This process stops all user sessions currently in progress and shuts down the MAPPER system.

Q

queue

To send output to an auxiliary device. For example, a report can be queued to a printer. *See also* requeue.

R

read password

See password.

real-time processing

A type of processing in which tasks are processed immediately by the system. Real-time tasks have a higher priority than ordinary batch or demand tasks. MAPPER software is real-time report processing software.

receiving label

A name or alphabetic field label into which the result of a MAPPER operation is stored. For example, a generated count of records in a report could be assigned to the COUNT receiving label.

receiving report

The report to which data is sent when using a manual function that processes two reports, such as the Add On (ADON) or Match (MA) functions. The receiving report is the lower mask of a double function mask. *Contrast with* issuing report. *See also* double function mask.

receiving variable

The first variable in an equation and the variable that contains the result of a calculation.

registration

A procedure done by the MAPPER system coordinator to add new users or runs to the system.

relational expression

A sequence of operands and relational operators used to compare values or character strings. When evaluated, it produces a value of 1 if the comparison is true or 0 if the comparison is false. *See also* Boolean logic, expression, relational operator.

relational operator

A special character used to compare values, such as =, >, <, <=, >=, or <> (equal to, greater than, less than, less than or equal to, greater than or equal to, and not equal to). *See also* Boolean logic, expression, relational expression.

remote run

A run that starts a run on another MAPPER system.

replacement string

A string of characters with which to replace the target string. *See also* target string.

report

The set of data that you work with in the MAPPER system. Reports are identified either by a unique report number or by a meaningful name you have given the report using the NAME run. *See also* report name.

report headings

Synonymous with heading lines.

report identifier (RID)

A specific report identified by a unique report number and drawer letter. For example, RID 2B refers to report 2 in drawer B.

report name

The characters used to refer to a report: either the report number and drawer letter or a meaningful name you have given the report using the NAME run. *See also* report.

report 0

A report that resides in each drawer in the MAPPER system. It serves as a template for the reports in its drawer when the Add Report function or run statement is used. *See also* drawer, format, function mask.

requeue

To send output already queued to an auxiliary device that failed to print the first time. *See also* queue.

restricted access function

A function requiring special permission from the MAPPER system coordinator before it can be used. Applies to functions that require advanced knowledge of MAPPER software or the operating system.

result

A temporary copy of data obtained by executing a manual function or run. It is held in scratch storage until released, duplicated, or replaced into a permanent report.

resumable function

A function that repeats or continues when the user presses **Resume**.

resume

To continue a function or run that was halted by a display. Press **Resume** to resume an operation.

retrieve

(1) To bring a MAPPER report that has been filed into a native file back into the MAPPER database. (2) To bring an earlier version of a MAPPER report back into the MAPPER database.

RID

See report identifier.

roll

To move vertically through the data in a report; to move a report forward or backward on the screen.

run

A series of instructions that the MAPPER system interprets to produce a report or perform other tasks such as updating reports. *See also* run control report, run statement.

run call

Synonymous with run name.

run control report

A MAPPER report containing sequential run statements of step-by-step instructions for processing reports, results, or other data. *See also* run, run statement.

run function call

An abbreviation used to request a function in a run statement (for example, SRH for Search). *See also* run statement.

run name

A name assigned to a run; used to access the run and execute it (for example, HELP). *Synonym for* run call.

run statement

A string composed of a run function call, options, fields, and subfields used to format an instruction that can be processed in a run. When the run statement is entered into a run control report, it can be processed to direct execution of MAPPER functions and other operations. *See also* run, run control report, run function call.

S

save flag

A date you place on line 2 starting at column 2 of a MAPPER report to prevent the report from being deleted before or on the date specified. Line 2 must be a period line in order to contain a save flag. Use the format @YYMMDD. *See also* period line.

SDF file
See System Data Format file.

search information lines
The lines displayed at the top of a result, showing how many lines were found by the search, the total number of lines searched, and the search parameters.

semicolon
A special character (;) used in some manual functions to separate equations.

shell
The user interface to the UNIX operating system that accepts and interprets commands.

sign off
To terminate MAPPER software at your terminal. The sign-on screen is displayed on the screen. *Contrast with* sign on. *See also* sign-on screen.

sign on
To initiate MAPPER software at your terminal by entering data that identifies you to the system. This data includes your user-id, department number, and password (if applicable). The active screen is displayed on your screen. *Contrast with* sign off. *See also* active screen.

sign-on cabinet
The cabinet a user enters automatically when signing on to the MAPPER system and in which the user primarily works. It is set by the MAPPER system coordinator as part of user registration. *See also* user registration.

sign-on password
See password.

sign-on screen

The screen you see before signing on to the MAPPER system. It shows your system name, your station number, and the level of MAPPER software. The word sign-on in your sign-on screen shows your station is inactive. *Contrast with* active screen. *See also* sign off.

single-field label

A label that identifies one report field when using equations in mathematical functions. *Contrast with* multiple-field label.

site

MAPPER software can support multiple databases referred to as sites. *See also* default MAPPER site.

site configuration

The devices (terminals, printers, or other MAPPER systems) connected to the host computer. *See also* host computer.

site identifier

See terminal site identifier.

site letter

An alphabetic designator assigned to each MAPPER system.

SNA

See system network architecture.

SOE

See start-of-entry character.

special characters

The set of characters (such as] and *) on your keyboard that are not alphabetic or numeric characters.

special line

A line starting with any character except tab, asterisk, or period. It acts as a tab line although an alphabetic or numeric character appears in column 1. *Synonym for* nontab line. *See also* line type, line type designator, tab line.

start-of-entry (SOE) character
The character represented on the screen by this symbol: ♦. It is used with the SOE Update function and other edit functions. *See also* edit function.

station
A terminal, workstation, or PC on the MAPPER system.

station number
The unique identifier for your terminal on the MAPPER system.

station sign-on screen
See sign-on screen.

string
See character string.

subcumulation
To cumulate using the Totalize (TOT) function until a key field value changes so that the result yields two result fields. *See also* cumulation.

subtotal
A sum of data for groups of related lines.

system coordinator
See MAPPER system coordinator.

System Data Format (SDF) file
A file in OS 1100 standard data format.

system directory
Reports that contain data names for cabinets, drawers, and reports. These are updated using the NAME run when adding, changing, or deleting a data name. *See also* data name.

system logo
See active screen, sign-on screen.

system message
A message the MAPPER system displays on the top line of the screen to alert you to a possible or actual problem.

system network architecture (SNA)

The communications architecture that addresses functions such as layered communications protocols, enabling communication among diverse units.

system screen

See active screen, sign-on screen.

T

tab

To move the cursor from one tab character position to another.

tab character

(1) A special character in MAPPER reports indicating a tab position on the screen. In this manual, a tab character on the screen is usually represented by a vertical bar (|) or center dot (·) but may be configured for your terminal as another character. (2) A special character used to specify tab lines. In this manual, a tab character is represented by a quadrate (⎕).

tab line

A line beginning with a tab character, usually a data line. It is controlled by the tab positions and input edit codes of report 0 and can extend up to 256 characters. *See also* input edit code, line type, line type designator, report 0.

target list

(1) A list of function calls in the HELP run. (2) A list of target words in an issuing report or list of words specified when you execute a function. The Word Locate (WL) and Word Change (WC) functions and run statements use target lists. *See also* HELP run, issuing report.

target string

The character string to be located or changed by a MAPPER function. *See also* character string, replacement string.

terminal site identifier (SID)

A number that identifies a site to a communications line.

tilde

A special character (~) that, when combined with another character, specifies to the MAPPER system how you want your text manipulated. *See also* control characters.

time format

A format that defines how a time is to be displayed; for example, HHMMSS displays a time as 121500.

time input specifications

The codes used to define the format of a time to process.

time output specifications

The codes used to define the output format of a time.

toggle

To switch between two modes of operation using the same key. For example, pressing the key activates one mode; pressing the key again returns to the original mode.

trailer line

A line automatically attached to a data line when certain functions or run statements, such as the Totalize (TOT) function, are used. Asterisk and period type lines are trailer lines to tab lines. *See also* asterisk line, line type, period line, tab line.

transfer

To copy, exchange, read, record, store, transmit, transport, or write data.

transmit

To type information and press the **Transmit** key, sending the information to the host computer.

transparent character
A character that occupies a position in a string. When the string is compared to another character string, the transparent character allows all characters in that position to be accepted. For example, the $ in the character string A$C allows ABC and A1C to be accepted. *See also* character string.

truncate
To cut off or shorten; to exclude. For example, when using the Add On (ADON) function to add a report that has lines longer than those of the receiving report, the new lines are truncated.

U

Universal Terminal System (UTS)
A family of terminals and related hardware, for example, UTS 400.

UNIX
The operating system for the host computer that supports the U Series MAPPER system.

update control
The ability of MAPPER software to guarantee that only one user at a time may be updating a given report.

update password
See password.

update result
A result that can be used to replace or delete lines in the original report from which it was produced. *See also* result.

user input
Any information you are instructed to type. In this manual it is shown in color and in lowercase letters.

user registration

The process by which the MAPPER system coordinator enters into the user registration report a user-id, sign-on password, sign-on department number, sign-on language, and sign-on cabinet for each user. *See also* department, password, sign-on cabinet, user-id.

user-id

A unique code assigned to each MAPPER software user for security reasons. User-ids are set up by the MAPPER system coordinator to allow access to certain functions and runs. *See also* user registration.

usercode

An identification code used to establish user identity on the A Series system, control security, and provide for segregation of files. *See also* log on.

UTS

See Universal Terminal System.

V

value label

A name of up to six characters that identifies a single numeric or text value in mathematical functions. *See also* constant label, field label.

vertical bar

A special character (|) that represents a tab character in MAPPER reports. On some terminals, the tab character is displayed as a space. *See also* tab character.

vertical operation

A summary calculation (for example, sum or average) performed on a single field across all data lines. *See also* horizontal operation.

vertical operator

A special character (such as + or /) that identifies the type of arithmetic operation to be performed on all values in a specified field.

vertical summation

The process of adding fields of data in a report and listing the totals at the end of the result. It is used with the Totalize (TOT) function.

W

WFL

See work flow language.

work flow language (WFL)

(1) A language used for constructing jobs that compile and run programs. WFL includes variables, expressions, and flow-of-control statements that offer the programmer a wide range of capabilities with regard to task control. (2) A language used to write jobs that control the flow of programs and tasks on the operating system.

workstation

See station.

write password

See password.

Index

Help Us To Help You

Publication Title

Form Number Date

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

□ Addition □ Deletion □ Revision □ Error

Comments

Name

Title Company

Address (Street, City, State, Zip)

Telephone Number

Help Us To Help You

Publication Title

Form Number Date

Unisys Corporation is interested in your comments and suggestions regarding this manual. We will use them to improve the quality of your Product Information. Please check type of suggestion:

□ Addition □ Deletion □ Revision □ Error

Comments

Name

Title Company

Address (Street, City, State, Zip)

Telephone Number

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1145 ST. PAUL MN 55164

POSTAGE WILL BE PAID BY ADDRESSEE

Unisys Corporation
Attn: MAPPER® Product Information
P.O. Box 64942 M.S.: 4792
St. Paul, MN 55164-0942 USA

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1145 ST. PAUL MN 55164

POSTAGE WILL BE PAID BY ADDRESSEE

Unisys Corporation
Attn: MAPPER® Product Information
P.O. Box 64942 M.S.: 4792
St. Paul, MN 55164-0942 USA