

UNISYS

A Series

Menu-Assisted Resource
Control (MARC)

**Operations
Guide**

Priced Item

July 1992
Printed in U S America
8600 0403-100

UNISYS

A Series

Menu-Assisted Resource
Control (MARC)

**Operations
Guide**

Copyright © 1992 Unisys Corporation
All Rights Reserved
Unisys is a registered trademark of Unisys Corporation.

Release Mark 4.0.0

Priced Item

July 1992
Printed in U S America
8600 0403-100

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 THIS DOCUMENT. Any product or related information described herein is only furnished pursuant and subject to the terms and conditions of a duly executed agreement to purchase or lease equipment or to license software. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such 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, 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.

RESTRICTED RIGHTS LEGEND. Use, reproduction, or disclosure is subject to the restrictions set forth in DFARS 252.227-7013 and FAR 52.227-14 for commercial computer software.

Correspondence regarding this publication may be forwarded using the Product Information card at the back of the manual, or may be addressed directly to Unisys, Technical Publications, 25725 Jeronimo Road, Mission Viejo, CA 92691-2792.

Page Status

Page	Issue
iii	-100
iv	Blank
v through xi	-100
xii	Blank
xiii through xxi	-100
xxii	Blank
xxiii	-100
xxiv	Blank
1-1 through 1-6	-100
2-1 through 2-19	-100
2-20	Blank
3-1 through 3-8	-100
4-1 through 4-20	-100
5-1 through 5-4	-100
6-1 through 6-54	-100
7-1 through 7-8	-100
8-1 through 8-33	-100
8-34	Blank
9-1 through 9-17	-100
9-18	Blank
A-1 through A-3	-100
A-4	Blank
B-1 through B-7	-100
B-8	Blank
C-1 through C-14	-100
D-1 through D-5	-100
D-6	Blank
E-1 through E-8	-100
Glossary-1 through 17	-100
Glossary-18	Blank
Bibliography-1 through 3	-100
Bibliography-4	Blank
Index-1 through 15	-100
Index-16	Blank

Unisys uses an 11-digit document numbering system. The suffix of the document number (1234 5678-xyz) indicates the document level. The first digit of the suffix (x) designates a revision level; the second digit (y) designates an update level. For example, the first release of a document has a suffix of -000. A suffix of -130 designates the third update to revision 1. The third digit (z) is used to indicate an errata for a particular level and is not reflected in the page status summary.

About This Guide

Purpose

Menu-Assisted Resource Control (MARC) is a menu-driven interface to Unisys A Series systems that also permits the direct entry of commands. MARC is designed to enable you to perform actions on the system without requiring that you have any knowledge of Unisys system commands.

The *A Series Menu-Assisted Resource Control (MARC) Operations Guide* leads the first-time user through a MARC session, and gives more experienced users the information needed to expand and modify MARC for their particular needs.

Scope

The *A Series Menu-Assisted Resource Control (MARC) Operations Guide* provides a brief overview of MARC, a description of the menu structure and how to traverse it, an explanation of how commands are generated and how the menu structure can be bypassed, and an outline of how MARC can be expanded or modified to suit your needs.

This document does not describe all the individual screens that are displayed in MARC. Documentation for specific screens is provided through the online help capability of MARC, as described in this guide.

Audience

This guide addresses a wide audience – from the user with very little data processing experience to the very experienced user.

Prerequisites

Any user familiar with his or her terminal can read this guide and operate MARC. The information in this guide along with the online help text enables you to use the product with no previous MARC experience.

How to Use This Guide

If you are new to MARC, read the beginning of the guide through Section 3, “Using the Online Help Facilities,” for information about how to conduct a basic MARC session. Experienced MARC users can read from Section 4 onward for information on more advanced topics.

Users who want to create customized versions of the MARC screens or to add customized help text can see Section 8, “Creating Custom Versions of MARC.” Users who want to define or use alternate default display languages, conventions, or character sets for a localized version of MARC can see Section 9, “Using MARC Internationally.”

Section 1, “Introducing MARC,” provides background information about MARC. When you have completed this introductory section, see the first part of Section 2, “Starting to Use MARC,” for instructions on how to log on to the system. Once you are logged on, you can complete Section 2 and Section 3, “Using the Online Help Facilities,” and try each of the features as they are described.

You need not have a terminal in front of you to understand this guide, because many illustrations of MARC screens are included in the text. These illustrations might vary somewhat from the actual screens in use at your installation.

Once you have finished Section 3, you will have a basic understanding of how to use MARC. However, you should read the later sections as well for information about some of the more sophisticated capabilities MARC provides.

This guide follows some conventions to make it easier to read.

- Unless otherwise noted, manuals referred to in the text are for A Series machines.
- Any acronyms that are not spelled out in the text can be found in the Glossary at the end of this guide. Many of the terms defined in the Glossary are shown in *italics* when they are first discussed in the text.
- This guide presents syntax diagrams in the form of *railroad diagrams*. For information on interpreting these diagrams, see Appendix E, “Understanding Railroad Diagrams.”
- This guide presents the names of specific keys on the terminal keyboard in uppercase characters, as in *press the TAB key*.

Organization

This guide is organized so that the minimum information you need to run MARC for the first time is located at the beginning of the guide, and later sections cover progressively more advanced topics.

In addition to the following sections, a glossary, a bibliography, and an index appear at the back of this guide.

Section 1. Introducing MARC

This section explains how MARC provides menus, forms, and help text to accomplish system operations.

Section 2. Starting to Use MARC

This section covers basic procedures for logging on, for logging off at the end of a session, and for accessing MARC from a system control terminal (SCT) or operator display terminal (ODT). The section also explains the layout of the various MARC screens, and how to use them.

Section 3. Using the Online Help Facilities

This section describes the various types of online text information that are available in MARC. It also provides instructions for accessing online help.

Section 4. Running Programs from MARC

This section describes how to run a program using MARC.

Section 5. Using COMS Windows and Dialogues

This section describes how MARC can make use of the features of the Communications Management System (COMS) windows and dialogues.

Section 6. Using Commands

This section describes how to use commands with MARC. The section describes the use of those commands that are specific to MARC, as well as the use of COMS commands, system commands, Print System commands, Work Flow Language (WFL) statements entered as commands, and commands to run system utility programs. The section also describes how to use the MARC option LOGOFFSTAT to control the display of session resources used during processing.

Section 7. Controlling System Security

This section introduces the various security categories of MARC users: nonprivileged, control capable, privileged, SYSTEMUSER, COMMANDCAPABLE, super user, and security administrator. The section also provides information on how the system administrator can use MARC to control system security.

Section 8. Creating Custom Versions of MARC

This section provides system administrators and programmers with the information they need to customize the standard screens provided with MARC, to add new screens, to use multiple versions of the screens, to add custom help text, to use COMS processing items with MARC, to add MARC directive commands, and to improve MARC performance.

Section 9. Using MARC Internationally

This section describes how to select a number of options relating to the use of localized versions of MARC, if one or more such versions of MARC have been provided for your use. This section shows system administrators how to use MARC to select system default versions of the display language, the convention, and the coded character set. The section also shows end users how to use MARC to override these default settings for an individual session. The section also contains information on translating MARC screens and help text.

Appendix A. Handling Faults in MARC

This appendix explains what happens at the ODT and at your terminal when MARC encounters a fault. The appendix also provides instructions for handling a fault.

Appendix B. Installing MARC on Your System

This appendix provides information on the installation process for MARC. It covers the use of the Simple Installation (SI) program and the alternative of manual installation, and provides details about configuring various terminal types for use with MARC. The appendix also provides instructions for initializing MARC and discusses how MARC can process messages produced by your installation's billing utility.

Appendix C. Commands Available in MARC

This appendix lists the names of all the commands that can be entered in the *Action* field of a MARC screen. The commands are grouped into MARC commands, COMS commands, system commands, Print System commands, and WFL statements entered as commands.

Appendix D. Menu Selections for Common Tasks

This appendix lists common operational tasks and shows the MARC menu selections that are available to perform these tasks.

Appendix E. Understanding Railroad Diagrams

This appendix explains how to read the syntax diagrams that are used in this guide.

Related Product Information

The following Unisys documents provide further information on topics mentioned in this guide. Other documents that contain information of interest are listed in the Bibliography at the back of this guide.

A Series CANDE Operations Reference Manual (8600 1500)

This manual describes how CANDE operates to permit generalized file preparation and updating in an interactive, terminal-oriented environment. This manual is written for a wide range of computer users who work with text and program files.

A Series Communications Management System (COMS) Configuration Guide (8600 0312)

This guide provides an overview of the basic concepts and functions of COMS. It includes instructions for creating a working COMS configuration and information on how to monitor and fine-tune COMS system performance. This guide is written for installation analysts, systems analysts, programmers, administrators, and performance analysts.

A Series Communications Management System (COMS) Operations Guide (8600 0833)

This guide explains how to perform terminal-based COMS functional tasks and serves as a reference to COMS commands. Syntax diagrams of COMS commands are provided with explanations and examples of how the commands can be used. This guide is written for terminal operators and computer operators.

A Series Communications Management System (COMS) Programming Guide (8600 0650)

The guide explains how to write online, interactive, and batch application programs that run under COMS. This guide is written for experienced applications programmers with knowledge of data communication subsystems.

A Series File Attributes Programming Reference Manual (8600 0064)

This manual contains information about each file attribute and each direct I/O buffer attribute. The manual is written for programmers and operations personnel who need to understand the functionality of a given attribute. The *A Series I/O Subsystem Programming Guide* is a companion manual.

A Series Help Utility Operations Guide (8600 0510)

This guide describes how to embed Help Utility commands in text to create a help book source file and how to process this source file with the Help Utility. This guide is written for inexperienced and experienced computer users, and writers or translators of online help text.

A Series Interactive Menugraph Generator (IMG) Operations Guide (8600 0411)

This guide explains how to use and run IMG, a software tool for the design and modification of Menu-Assisted Resource Control (MARC), Interactive Datacomm Configurator (IDC), BNA Version 1, BNA Version 2, and IMG screens and forms. This guide is written for users who want to customize these screens for their system.

A Series Message Translation Utility (MSGTRANS) Operations Guide (8600 0106)

This guide describes how to use the Message Translation Utility (MSGTRANS) to translate compiled program messages from any natural language to any other natural language. It provides complete instructions for running and using the screen and batch interfaces of MSGTRANS. This guide is written for programmers and translators who create and translate program messages in a MultiLingual System (MLS) environment.

About This Guide

A Series Micro A System Installation Guide (5044100)

This guide describes Micro A hardware and software installation. It also provides information on hardware components, basic operation considerations, and troubleshooting. This guide is written for Micro A users.

A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide (8600 0288)

This guide describes how to use the MLS environment, which encompasses many Unisys products. The MLS environment includes a collection of operating system features, productivity tools, utilities, and compiler extensions. The guide explains how these products are used to create application systems tailored to meet the needs of users in a multilingual or multicultural business environment. It explains, for example, the procedures for translating system and application output messages, help text, and user interface screens from one natural language to one or more other languages; for instance, from English to French and Spanish. This guide is written for international vendors, branch systems personnel, system managers, programmers, and customers who wish to create customized application systems.

A Series PCMARC Installation and Operations Guide (8600 0817)

This guide describes how to install and operate PCMARC. PCMARC presents Menu-Assisted Resource Control (MARC) screens on a workstation connected to an A Series host that is running MARC. PCMARC includes windowing environment software, a terminal emulator for your workstation, and a communication interface for handling PCMARC communication with the host. This guide is written for A Series system operators.

A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide (8600 1039)

This guide describes the features of the Print System and provides a complete description of its command syntax. This guide is written for programmers, operators, system administrators, and other interactive users of Menu-Assisted Resource Control (MARC) and CANDE.

A Series Printing Utilities Operations Guide (8600 0692)

This guide describes how to use the Print System utilities: Backup Processor, SYSTEM/BACKUP, and LTTABLEGEN. This guide is written for programmers, system administrators, and interactive users of Menu-Assisted Resource Control (MARC) and CANDE who are familiar with the concepts and use of the Print System.

A Series Security Features Operations and Programming Guide (8600 0528)

This guide describes the security features available to users and provides instructions for their use. This guide is written for users who are responsible for maintaining the security of their individual programs and data.

A Series System Commands Operations Reference Manual (8600 0395)

This manual gives a complete description of the system commands used to control system resources and work flow. This manual is written for systems operators and administrators.

A Series System Messages Support Reference Manual (8600 0429)

This manual presents operating system error messages and explains the most likely cause of each error and the most effective response to each message. This manual is written for operators and programmers responsible for the operation of A Series systems, and for the resolution of error conditions on those systems.

A Series System Software Support Reference Manual (8600 0478)

This manual describes a number of facilities used for system monitoring and debugging, including BARS, DUMPANALYZER, LOGANALYZER, and LOGGER. It also describes the format of the SUMLOG file. This manual is written for system support personnel and operators.

A Series Systems Functional Overview (8600 0353)

This manual presents an overview of the A Series systems and serves as a central source of information for these systems. This overview is written for both new and experienced users of A Series systems, and for anyone wanting an introduction to these systems.

A Series Work Flow Language (WFL) Programming Reference Manual (8600 1047)

This manual presents the complete syntax and semantics of WFL. WFL is used to construct jobs that compile or run programs written in other languages and that perform library maintenance such as copying files. This manual is written for individuals who have some experience with programming in a block-structured language such as ALGOL and who know how to create and edit files using CANDE or the Editor.

Contents

About This Guide	v
Section 1. Introducing MARC	
Anyone Can Use MARC	1-1
You Can Use MARC at Any Terminal	1-1
User Terminals	1-2
System Control Terminals	1-2
What MARC Can Do for You	1-2
Menu Mode and Command Mode	1-4
Menu Mode	1-4
Command Mode	1-5
MARC Security Features	1-5
Customized Screens, Help Text, and Commands	1-6
Customized Versions of MARC Screens	1-6
Customized Versions of MARC Help Text	1-6
Custom Commands	1-6
Localized Versions of MARC	1-6
Section 2. Starting to Use MARC	
Logging On	2-1
Usercodes and Passwords	2-2
Accesscodes and Chargecodes	2-3
If the LOGON Screen Does Not Appear	2-3
Recording of Logon Records in the System Log File ..	2-4
Screens	2-4
Input Fields	2-4
Screen Layout	2-5
Refreshing the Screen	2-7
Transmitting Input	2-8
Action Field Typeahead	2-8
Finding Your Way through the Screens	2-9
Using Menus	2-10
Menu Selections	2-10
Using a Mouse to Make Menu Selections	2-11
Using Forms	2-11
Form Input Fields	2-12
Command Generation	2-13
Using the Action Field	2-13
Actions Available on Most Screens	2-13
HOMe	2-13
PRev	2-13
GO	2-14
PArEnt	2-14

COmnd	2-14
KEys	2-14
REturn	2-15
Plus (+)	2-15
Minus (-)	2-15
Actions Available Only with File Lists	2-15
First	2-15
LAsT	2-16
SEQ	2-16
COL	2-16
Using the Choice Field	2-16
Choice Field Typeahead	2-16
Command Generation	2-17
Displaying Output from Commands	2-17
Continued Information	2-18
End of Output	2-18
Logging Off	2-18
Using BYE	2-18
Using HELLO	2-19
Using SPLIT	2-19
Using ?CLOSE	2-19

Section 3. Using the Online Help Facilities

Short Help Text	3-1
Long Help Text	3-2
Online Definitions of Terms	3-3
Using the HELP Command	3-5
How to Enter HELP	3-5
Finding Keywords	3-6
Finding Reserved Keywords	3-6
Selecting Other Help Books	3-6
Using the TEACH Menu	3-6
Using a Mouse to Display Help Text	3-8

Section 4. Running Programs from MARC

Distinguishing between a Task and a Job	4-1
Synchronous Tasks	4-2
Asynchronous Jobs	4-2
Starting a WFL Job	4-2
Starting a WFL Job from the START Screen	4-3
Starting a WFL Job from the Action Field	4-4
Running a Task	4-5
Running a Task from the RUN Screen	4-5
Defining the Task to Be Run	4-5
Making File Equations	4-7
Making Task Attribute Assignments	4-8
System Switch Values	4-8
Program Dump Options	4-8

Other Options	4-9
Attributes Affecting Testing, LSNs, Printing, and Language	4-10
Running a Task from the Action Field	4-10
Displaying the TASKSTATUS Screen	4-11
Screen Display	4-12
Entries in the Action Field	4-12
Available Actions	4-13
System Command Functions Available as Actions	4-13
Displaying the TASKVIEW Screen	4-14
Displaying the Task Window	4-15
Standard Tasks	4-16
Ended Tasks	4-16
Remote Window	4-16
Monitoring the Progress of a Task	4-17
Task Initiation	4-17
Entering Tasking Mode	4-17
TASK Action	4-18
VIEW Action	4-18
?MARC Command	4-18
Exiting Tasking Mode	4-18
Task Suspension	4-19
Task Termination	4-19
Providing a Final Task Message	4-20

Section 5. Using COMS Windows and Dialogues

Entering COMS Commands	5-1
Using Multiple MARC Dialogues	5-2
Switching between Windows and Dialogues	5-2
How MARC Stores Messages	5-3
How a Dialogue Inherits Attributes	5-4

Section 6. Using Commands

Entering Commands	6-1
Commands Available in MARC	6-2
MARC Commands	6-3
ALLMSG (All Messages)	6-3
BDIR (Backup Directory)	6-4
BP (Breakpoint)	6-4
?CONTINUE (Continue MARC Processing)	6-5
DIRECTIVE (Define Directives)	6-5
FAMILY (Set Family Specification)	6-5
FILES (Display File Titles)	6-7
GENERATE PASSWORD	6-9
HELP (Display Help Text)	6-9
LANGUAGE (Change Language Specification)	6-10
Effects of the LANGUAGE Command on Messages and the Menugraph	6-12

Contents

Effects of the LANGUAGE Command on Screens	6-13
LIST (List a File)	6-13
Additional Options of the LIST Command	6-14
Displaying Printer Backup Files	6-16
MENU (Update Menugraph Version)	6-17
MSC (Message Control)	6-17
Understanding How the System Uses MSC Commands	6-21
Entering MSC Commands	6-22
Displaying Current MSC Commands	6-22
Suppressing and Showing Messages	6-22
Combining Commands for Unique Message Control	6-26
Creating and Using MSC Command Files	6-27
MSC Command Examples	6-29
MSG (Messages)	6-30
NEW (Update Menugraph or Book Version)	6-31
NEWS (Display News File)	6-32
PASSWORD	6-33
PDIR (Process Directory)	6-33
SHOW PASSWORD	6-39
SMSG (System Messages)	6-40
SUSPENDUSERCODE (Suspend Usercode)	6-40
TEACH (Display Help Text)	6-41
WRU	6-41
COMS Commands	6-42
System Commands	6-42
File and Message Handling Commands	6-43
Primitive Commands	6-43
WFL Statements Entered as Commands	6-44
Print System Commands	6-45
Controlling Session Print Attributes	6-45
DESTNAME Command	6-45
JOBSUMMARY Command	6-46
JOBSUMMARYTITLE Command	6-47
NOJOBSUMMARYIO Command	6-47
PRINTDEFAULTS Command	6-48
System Utilities	6-49
System Administrator Utilities	6-49
Debugging Utilities	6-50
General Utilities	6-50
Using the LOGOFFSTAT Option	6-52

Section 7. Controlling System Security

Understanding Security Categories	7-1
Usercode Security Categories	7-1
Security Administrator	7-1
SYSTEMUSER	7-2
Control Capable	7-2

Privileged User	7-2
Other Security Categories	7-3
Super User	7-3
COMMANDCAPABLE	7-3
Restricting Access to MARC Screens	7-4
Setting the Security Privileges	7-5
Using COMS to Set Privileges	7-5
Changing the USERDATAFILE Usercode Database	7-5
Changing the Password for a Usercode or Accesscode	7-6
Writing Local Security Procedures	7-6
SECURITYSUPPORT Library	7-6
Excessive Log-on Attempts	7-6
Understanding the Default Procedure	7-7
Modifying the Maximum Number of Log-on Attempts	7-7
Accessing Security and Help Books	7-8

Section 8. Creating Custom Versions of MARC

Customizing the MARC Menugraph	8-1
Changing the Screens	8-2
Customizing Menus	8-2
Customizing Forms	8-3
Using the Message Translation Utility (MSGTRANS)	8-3
Setting the Usercode Default Home Menu	8-3
Making a Custom Menugraph Available to Users	8-4
Leftover Screens	8-5
Who May Update a Menugraph	8-5
Error Messages	8-5
Using Multiple MARC Menugraphs	8-6
Active Menugraph List	8-6
Substitute Menugraph	8-7
Usercode Default Menugraph	8-7
MARC Session Language Menugraph	8-8
System Default Menugraph	8-8
English Language Menugraph	8-9
Displaying the List of Active Menugraphs	8-9
Declaring a Substitute Menugraph	8-10
Bypassing the Menugraph Search Order	8-11
Customizing the Help Text	8-11
Writing Your Own Help Text	8-12
Making Custom Help Text Available to Users	8-12
Using COMS Processing Items with MARC	8-13
MARCINPUT and MARCOUTPUT Agendas	8-13
Formatting Display Output	8-13
Guidelines for Writing Processing Items	8-13
Using Directive Commands	8-14
DIRECTIVE Command	8-15
Directives Definition File	8-18
Writing Directive Commands	8-18

Format of the Source Code	8-19
Parameters	8-19
COMS_HEADER Parameter	8-20
COMMAND Parameter	8-20
TAG Parameter	8-20
DIRECTIVE Parameter	8-21
GET_DIALOG_ATTRIBUTE Parameter	8-21
RESPONSE_FORMATTER Parameter	8-23
SCRATCH Parameter	8-23
Procedure Result	8-24
Sample Directive Command	8-25
Improving MARC Performance	8-32
Increasing the Number of Concurrent MARC Copies	8-32
Moving Internal Files	8-33

Section 9. Using MARC Internationally

Setting Options to Support Localized Versions of MARC	9-1
Choosing a Ccsversion	9-2
Understanding the Elements of a Ccsversion	9-2
Setting the Ccsversion (System Level Only)	9-3
Choosing a Language	9-4
Setting the System Language	9-5
Setting the System Language in MARC	9-5
How the System Language Affects MARC	9-5
Setting the MARC Session Language	9-6
Setting the COMS Application Session Language	9-7
Setting the Language at the Usercode Level	9-7
Setting the Language for a Task	9-8
How MARC Determines the Language to Use for Messages	9-8
Choosing a Convention	9-8
Setting the System Convention	9-9
Setting the Session Convention	9-10
Setting the Convention at the Usercode Level	9-11
Setting the Convention for a Task	9-11
Creating Localized Versions of MARC	9-11
Localizing MARC Menus and Forms	9-11
Generating the Localized MARC Menugraph File	9-11
Positioning the File in the Active Menugraph List	9-12
Localizing the MARC Online Documentation	9-12
Creating Translated Help Text	9-13
Associating the Translated Text with the Localized Menugraph File	9-13
Localizing Help Text Keywords	9-13
Localizing MARC Messages	9-14
Storing Localized Versions of Other MARC Screen Elements	9-17

Appendix A. Handling Faults in MARC

What Happens at the ODT	A-1
What Happens at Your Terminal	A-2
What You Can Do	A-2

Appendix B. Installing MARC on Your System

Installing the MARC Software	B-1
Files You May Omit	B-2
System Libraries	B-2
Configuring the ODT for Data Comm Mode	B-2
Requirements for Data Comm Mode	B-3
Using Two Modes on Your System	B-3
Switching between Modes	B-4
Limitations	B-4
Commands for Data Comm Mode	B-4
??MARC (Transfer to Menu-Assisted Resource Control) Primitive Command	B-4
??ODT (Return from Menu-Assisted Resource Control) Primitive Command	B-4
SM (Send to MCS) System Command	B-5
Configuring User Terminals	B-5
Configuring Terminals in Forms Mode or Single-Line Mode	B-6
Configuring Terminals for Full-Screen Transmission	B-6
Initializing MARC	B-7
Reviewing Messages Produced by Your Billing Utility	B-7

Appendix C. Commands Available in MARC

MARC Commands	C-1
COMS Commands	C-2
System Commands	C-3
Print System Commands	C-10
WFL Statements Entered as Commands	C-12

Appendix D. Menu Selections for Common Tasks

Appendix E. Understanding Railroad Diagrams

What Are Railroad Diagrams?	E-1
Constants and Variables	E-2
Constraints	E-2
Following the Paths of a Railroad Diagram	E-5
Railroad Diagram Examples with Sample Input	E-6

Contents

Glossary	1
Bibliography	1
Index	1

Figures

1-1.	MARC Home Menu	1-4
2-1.	MARC LOGON Screen	2-1
2-2.	Sample Screen Layout	2-5
2-3.	Sample MARC Menu	2-10
2-4.	Sample MARC Form	2-12
2-5.	Sample OUTPUT Screen	2-17
3-1.	Sample Short Help Text Display	3-2
3-2.	Sample Long Help Text Display	3-3
3-3.	Sample Online Glossary Definition	3-4
3-4.	TEACH Menu	3-7
4-1.	START Screen	4-3
4-2.	START Screen (Filled)	4-4
4-3.	RUN Screen	4-6
4-4.	FILEEQUATE Screen	4-7
4-5.	TASKATTR Screen	4-8
4-6.	TASKSTATUS Screen	4-11
4-7.	TASKVIEW Screen	4-15
6-1.	COMND Screen	6-2
6-2.	Logoff Statistic Screen	6-53
9-1.	MSGTRANS Start Session Screen	9-15
9-2.	MSGTRANS Home Screen	9-16
E-1.	Railroad Constraints	E-5

Tables

2-1.	Elements Common to MARC Screens	2-6
6-1.	LANGUAGE Command Formats	6-11
6-2.	LANGUAGE Command Goals	6-12
6-3.	MSC Command Message Type Specifications	6-24
6-4.	Selected Message IDs	6-24
6-5.	Message ID FROM Qualifier Values	6-25
6-6.	MSC Command Message Destination Specifications	6-26
6-7.	PDIR Command Item Names (SHOW, SORT, SELECT Parameters)	6-35
8-1.	Conversation Area of Message Headers	8-14
8-2.	GET_ATTRIBUTE Procedure Result Values	8-21
8-3.	GET_ATTRIBUTE Procedure Result Formats	8-21
8-4.	EXTERNALCOMMAND Library Procedure Result Values	8-24
B-1.	Sample Billing Utility Bill Processing Schedule	B-7
C-1.	MARC Commands	C-1
C-2.	COMS Commands Available in MARC	C-2
C-3.	System Commands Available in MARC	C-3
C-4.	Primitive Commands Available in MARC	C-10
C-5.	Print System Commands Available in MARC	C-10
C-6.	WFL Statements That Can Be Entered as Commands in MARC	C-12
D-1.	Menu Selections for Common Tasks	D-1

Section 1

Introducing MARC

Menu-Assisted Resource Control (MARC) is a menu-driven interface to A Series computer systems. That is, MARC is system software that enables you to use menus to control operations on your A Series system. MARC runs in conjunction with the Communications Management System (COMS), and is available at user terminals, which are also known as *data comm terminals*.

Instead of having to remember system commands and their exact syntax, you can simply pick a *selection key* from a MARC *menu* to perform a system operation, such as running a program or copying a file. A menu selection either executes a command directly, or it leads to a *form* that you fill in with the details of what you want to do.

As you become more familiar with the syntax of commands used on A Series systems, you might prefer to use these commands instead of choosing menu selections. You can enter commands in the *Action* field of a MARC screen. MARC supports most system commands, all the control commands for COMS, all Work Flow Language (WFL) statements entered as commands, and a number of system utilities.

Because MARC runs as a *window* under COMS, you can have up to eight separate MARC sessions (known as *dialogues*) running at your terminal simultaneously. The eight MARC sessions are labeled MARC/1 through MARC/8 on the screen. You can move at any time from MARC to the Command and Edit (CANDE) window that is available in COMS. The CANDE window also provides up to eight simultaneous CANDE dialogues at your terminal. The CANDE window provides access to all the capabilities of the CANDE message control system (MCS).

Anyone Can Use MARC

MARC is suitable for any A Series user. For new users and others who do not need a great deal of control over the system, MARC provides simple menus with only the essential functions. For experienced users such as system operators, MARC provides menus with an extensive range of selections covering every aspect of system operations. The system administrator decides whether a user needs the simpler set of menus or a more advanced set.

You Can Use MARC at Any Terminal

You can use MARC no matter what type of terminal you have, as long as the terminal is connected to an A Series system running the MARC software. As far as MARC is concerned, there are two main classes of terminal: user terminals and system control terminals (SCTs).

Introducing MARC

If your terminal is a Unisys personal computer (PC) or a compatible PC that supports the PCMARC user interface, see the *A Series PCMARC Installation and Operations Guide* for information on the use of PCMARC.

User Terminals

User terminals are those terminals from which end users can communicate with the system. User terminals are of many different hardware types, including the following:

- Unisys personal computers (PCs)
- BTOS workstations
- T25 and T27 terminals
- ET 1100 and ET 2000 series workstations
- TD terminals

Some hardware types support *scrolling*, a mode of operation in which display output can appear to move upward as new lines are added at the bottom. Generally speaking, you can obtain best results in MARC if you turn scrolling off at your terminal.

For more information about how to configure your terminal for MARC, see “Configuring User Terminals” in Appendix B, “Installing MARC on Your System.”

System Control Terminals

On an A Series system, the system control terminal (SCT) is the terminal the system operator uses to communicate directly with the operating system or maintenance processor. When the SCT is communicating with the operating system, the terminal is called the *operator display terminal* (ODT). To use MARC, the ODT must first be set to operate in *data comm mode*. See “Configuring the ODT for Data Comm Mode” in Appendix B, “Installing MARC on Your System,” for more information.

What MARC Can Do for You

If your primary goal in using an A Series system is to run one or more application programs, such as a database inquiry or payroll program, you might use only a few of the MARC menus. Your system administrator might have created a custom MARC menu containing only those MARC features that you use, together with selections that run your application programs.

Even if you use MARC primarily to run application programs, you might find other MARC capabilities useful. For example, you can use MARC to control the printing of reports that your programs produce.

Other tasks for which you can use MARC include

- Performing file maintenance functions such as copying, removing, or retitling files; displaying file titles; and displaying the contents of your disk files.
- Initiating programs such as system utilities and your application programs. You can submit batch or online programs, and you can interact directly with the online programs. While the program is still running, you can switch to a screen that shows the status of the program being run, or you can switch to other screens that enable you to perform other system operations.

If you are a system operator or system administrator with the corresponding security privileges defined for your usercode, the full range of MARC menus and commands is available to you. In addition to the tasks just discussed, you can use MARC for more advanced operations such as

- Monitoring and controlling all jobs and tasks on the system
- Monitoring and controlling the status of system hardware resources, including all peripherals
- Controlling the Print System (PrintS) and Remote Print System (ReprintS)
- Controlling the data comm network, including all message control systems (MCSs) such as COMS
- Controlling multihost networks, including the following types of networks:
 - BNA Version 1
 - BNA Version 2
 - Open Systems Interconnection (OSI)
 - Systems Network Architecture (SNA)
 - Transmission Control Protocol/Internet Protocol (TCP/IP)
 - X.25

If your installation has created a billing utility using the interfaces and functions provided by Unisys, MARC can supply information to, or display information returned from, that utility. For example, MARC can request that bill processing be performed whenever any of the following events occur:

- Beginning of session (BOS)
- End of session (EOS)
- Beginning of task (BOT)
- End of task (EOT)
- Session or task status inquiry

MARC can display billing messages whenever some of these events occur. For more information, see “Reviewing Messages Produced by Your Billing Utility” in Appendix B, “Installing MARC on Your System.” For information about the A Series billing support facilities, see the *A Series System Administration Guide*.

Menu Mode and Command Mode

You can use MARC in menu mode or in command mode. MARC presents the same screens in both modes – the difference is in the way you use these screens.

Menu Mode

In *menu mode*, you use the MARC menus that display selection keys corresponding to various operations functions. Selections on the MARC home menu correspond to general categories of operations functions. When you enter one of these selections, either MARC executes the corresponding system command immediately, or it displays another menu that breaks down the function in more detail.

In most cases, you need traverse no more than two or three menus to reach the most specific level. Menu selections at that level cause MARC to execute a system command on your behalf. If the command needs parameters, MARC prompts you for those values with a form.

Figure 1–1 illustrates the MARC home menu.

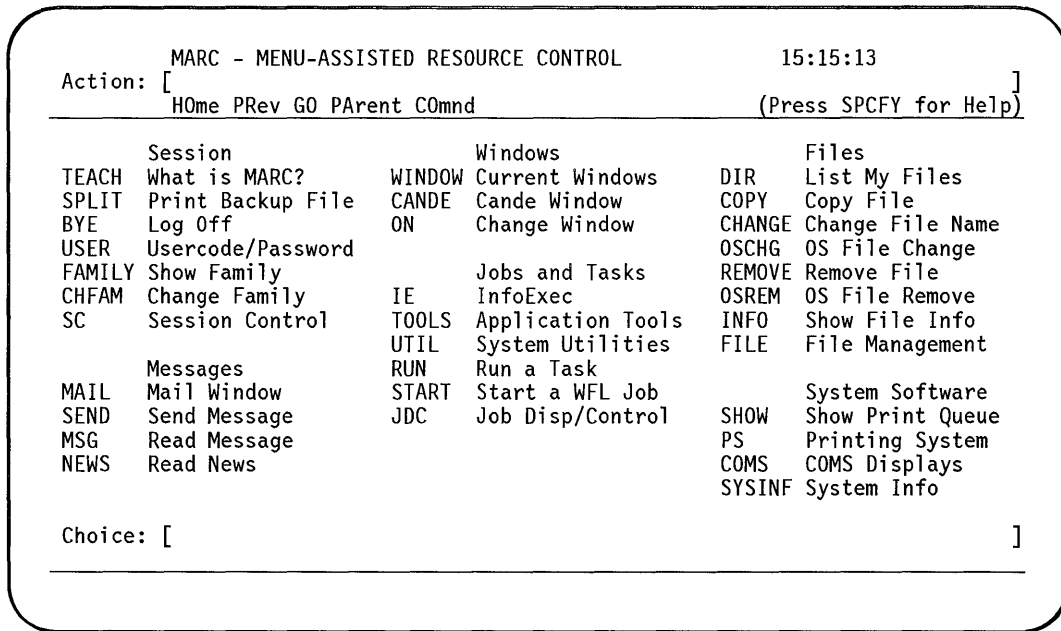


Figure 1–1. MARC Home Menu

At any point in this process, you can obtain online help information by pressing a single function key: the *SPCFY* (specify) key. Help is available for

- Each menu and form as a whole
- Each individual menu selection
- Each field on a form

In addition, MARC provides online definitions of terms used in the online help text. Within the help text, each term that has a glossary definition is highlighted. You can traverse to the glossary definition by pressing a single function key, and then return later to the text that you were reading. For more information about online help, see Section 3, “Using the Online Help Facilities.”

MARC also provides a TEACH selection on the home menu for users who are new to MARC. This selection leads you to online tutorials that introduce A Series systems and the MARC interface. It also provides access to the online definitions of terms used in the help text, to an online list of the published documentation for A Series systems, and to the online Master Glossary of A Series terms. For more information about online tutorials, see “Using the TEACH Menu” in Section 3.

As you become more familiar with A Series operations functions, you might want to execute operations functions directly, without going through the menus. MARC enables you to do so in either of two ways:

- Using the GO action to traverse to a particular menu without passing through the intermediate menus
- Using command mode, described next

Command Mode

In *command mode* you enter system commands directly. These are the commands that MARC would otherwise generate for you, based on your menu selections. You can enter system commands in the *Action* field of most menus and forms. MARC displays the output produced by the command as follows:

- If the command produces output that fits on one or two lines, MARC displays that output on the bottom two lines of the screen.
- If the command produces longer output, MARC displays an OUTPUT screen. See “Displaying Output from Commands” in Section 2, “Starting to Use MARC,” for information about the OUTPUT screen.

In addition to system commands, command mode enables you to enter MARC commands, COMS commands, Print System commands, and WFL statements entered as commands. See Section 6, “Using Commands,” for details.

MARC Security Features

MARC security features enable you to prevent unauthorized access to system functions. Each MARC user is required to log on by entering a usercode and password at the start of a MARC session. MARC examines the privileges that were defined for that usercode by the system administrator, and provides any of several levels of access to system commands. If a user does not have access rights to a particular command, MARC suppresses the corresponding menu selection from menus presented to that user and rejects attempts by that user to enter the command directly. For more information about MARC security features, see Section 7, “Controlling System Security.”

Customized Screens, Help Text, and Commands

You can create customized versions of the MARC interface to suit the needs of your installation and of individual users. You can create and modify customized versions of MARC screens and help text, and you can create and modify new commands.

Customized Versions of MARC Screens

The tool used for creating customized versions of MARC screens is the Interactive Menugraph Generator (IMG). IMG is an interactive tool that steps you through the process of creating, modifying, or deleting menus and forms. You can change the wording that appears on existing screens, you can add or delete menu selections, and you can change the wording of form fields, or even create new forms. For each menu selection, you can specify the system command it executes or the name of another menu it causes to be displayed. In addition, you can define entire linked sets of screens for specific users or groups of users. A linked set of screens designed using IMG is called a *menugraph*. The set of screens defined for MARC is known as the *MARC menugraph*.

Customized Versions of MARC Help Text

The tools used for creating customized versions of MARC help text are the A Series Editor and the A Series Help Utility. Through the Editor, you can modify existing help text and you can write additional help text for any new menus and menu selections you have added. You then use the Help Utility to convert the modified help text file into a file called a *help book*, which MARC can access directly.

The Help Utility enables you to create multiple help books. You can use IMG to specify the default help book to be used with each menugraph. In this way, you can make different sets of help text available to different users.

Custom Commands

MARC provides a *directive command* feature that enables you to create new commands and make them available to MARC users. You define the functions of these commands by writing a library containing ALGOL procedures. Thereafter, end users can enter these commands in MARC just as if they were MARC commands.

Localized Versions of MARC

A localized version of MARC is a version of MARC that has been adapted to meet the standards and functional requirements of a particular location. A localized version can display MARC screens and help text in natural languages other than English: French or Italian, for example. Localized versions can use alternate character sets and they can follow alternate conventions in displaying items such as dates and times.

In general, localized versions of MARC are created by Unisys. For installations wanting to create their own localized versions, appropriate tools and instructions are provided.

Section 2

Starting to Use MARC

Before you can use MARC on your terminal, the MARC software must be installed and initialized on your system, and the LOGON screen must be displayed. In most cases, the LOGON screen appears on the screen when you switch on your terminal and establish connection with the system. For more information, see Appendix B, "Installing MARC on Your System."

Logging On

You use the LOGON screen to identify yourself to MARC and the rest of your system. Figure 2-1 illustrates the MARC LOGON screen.

```
LOGON - Menu-Assisted Resource Control          03:36 PM
-----
Welcome.

Please enter your usercode [          ]
...and your password [          ]

You have access rights to the system without special
identification. If you want to access the system in this
way, enter a star * as your usercode.
-----
```

Figure 2-1. MARC LOGON Screen

If your system includes an Open Systems Interconnection – Message Handling System (OSI-MHS), MARC automatically checks to see if you have any new mail after you log on. If you do have new mail, a message is displayed on the status line. A message is also displayed if you get any new mail after you have logged on to MARC.

Usercodes and Passwords

A *usercode* is a unique name that identifies you to the system. A *password* is a character string that is associated with the usercode, and that is used to identify legitimate users of the usercode. To log on, type your usercode in the input field that requests it. Press the TAB key, and then type your password in the next field. Press the *XMIT* (transmit) key to display the MARC home menu.

If you do not have a usercode and password, contact your supervisor or the system administrator. You might be able to choose your own usercode, or you might have one assigned to you. Normally, you keep the same usercode for as long as you work with the system. Other users may know what your usercode is. Your password, however, should be known only to you and the system administrator.

Usercodes and passwords can have certain special features. The following paragraphs describe these features.

One-Time-Only Usercode

In some cases, you might be assigned a one-time-only usercode, which is good only for the duration of one session. Once you log off from that session using a one-time-only usercode, you need another usercode to log on again.

Secure-Video

On most Unisys terminals, the password field is presented in *secure-video mode*. This mode prevents your password from appearing on the screen as you enter it. If you fail to provide a correct password, you must reenter the usercode along with the password.

Password Aging

If you have been assigned a password that expires after a length of time, your password is known as an aging password. On an aging-password system, you might see the CHGPASS screen when you log on. If your password is due to expire, you are given a warning that indicates how many days are left for that password. If your password has already expired, you might need to contact the system administrator and arrange for a new password before you can log on to the system.

Limited Log-on Attempts

If you fail to enter a valid usercode and password in several consecutive attempts at logging on, the terminal enters a state known as *not ready*. As a result, you cannot use the terminal until the system administrator, or another user with the appropriate privileges, has readied that terminal.

By default, the number of log-on attempts permitted is 10, but the system administrator can change that value for your system to any number between 0 – to permit unlimited log-on attempts – to 15. See “Writing Local Security Procedures” in Section 7, “Controlling System Security,” for additional information on local security procedures.

Super-User Status

The message shown at the bottom of the screen in Figure 2-1 appears at an operator display terminal (ODT) or at a user terminal that has been marked as a *super-user-capable station* by means of the Communications Management System (COMS) Utility program. If your terminal is super-user-capable, you can log on by entering an asterisk (*) instead of a regular usercode and no password is required. For more information, see Section 7, “Controlling System Security.”

Accesscodes and Chargecodes

Depending on the security features in effect on your system, you might be required to enter—in addition to a usercode and password—an accesscode, an accesscode password, and a chargecode, as follows:

- An *accesscode* identifies you as having access rights to those files that are guarded by a special file known as a *guard file*. The accesscode itself might also have an associated password. If an accesscode is required, MARC displays another screen requesting the accesscode and accesscode password.
- A *chargecode* identifies you as a member of a charge group for accounting purposes. If a chargecode is required, MARC also displays a screen requesting the chargecode.

If a screen appears requesting an accesscode or chargecode and you decide you want to log on to another usercode, you can return to the LOGON screen by transmitting blank fields on the accesscode or chargecode screen. The LOGON screen appears also if you fail to enter a valid accesscode or chargecode in three consecutive attempts.

If the LOGON Screen Does Not Appear

On any terminal that is controlled by COMS, MARC normally presents the LOGON screen to the user as soon as the terminal is switched on and connected to the A Series system. However, the LOGON screen fails to appear in the following situations:

- The security administrator has used the COMS Utility to set the automatic log-on feature to a default usercode for the station.
See Section 5, “Using COMS Windows and Dialogues,” for more information on COMS windows.
- The terminal comes up in single-line mode.
Some older types of terminal use single-line mode, and cannot display menus or forms. In this case, the terminal displays “Please log on” or a similar message. From this point, the log-on procedure is the same as the procedure for logging on to the Command and Edit (CANDE) message control system (MCS). See “Configuring User Terminals” in Appendix B, “Installing MARC on Your System,” for additional information on single-line mode.
- The terminal is an ODT that is not in data comm mode.

In this case, use the ??MARC (Transfer to Menu-Assisted Resource Control) primitive system command to initiate data comm mode at your ODT before logging on to MARC. See “Configuring the ODT for Data Comm Mode” in Appendix B for more information.

Recording of Logon Records in the System Log File

The system maintains a record of activities such as session log-on and log-off, the beginning and ending of tasks, and unauthorized file access attempts in the *system log file*, a file called SYSTEM/SUMLOG.

When a user logs on to the system, MARC notes the event. By default, however, MARC postpones the actual writing of the logon record to the system log file until a second event occurs that concerns the usercode and warrants an entry in the log file. At that time, MARC writes both entries – that is, the logon record and the entry recording the second activity – to the log file. This delay in recording logon records helps speed the MARC initialization process.

Systems with the optional InfoGuard security enhancements can override the default MARC behavior regarding the recording of logon records. To specify that all logon records are to be written at the time each user logs on rather than later, the security administrator can use the SECOPT (Security Options) system command to assign a CLASS option value of S2 to the system. To override the default behavior for an individual usercode, the security administrator can grant that usercode either SECADMIN (security administrator) or PU (privileged user) privileges. The system writes logon records immediately for usercodes with either of these statuses, regardless of the value of the CLASS option. See the *A Series Security Administration Guide* for more information on the CLASS option and on SECADMIN and PU privileges.

Screens

The term *screen* in MARC refers to the contents of a predefined MARC display on a terminal. Some portions of MARC screens are set aside for you to enter information, and other portions present text designed to help you decide what to enter.

Some terminals display MARC screens in a windowing environment such as Microsoft® Windows™ or INFOVIEW™. In these cases, the MARC screen might show additional text, not directly related to MARC, at the top and bottom of the display.

Input Fields

The areas of the screen that are set aside for you to enter information are called *input fields*. Input fields are typically delineated on the screen by solid triangles. However, the delineating characters might vary, depending on what kind of terminal you are

Microsoft is a registered trademark of Microsoft Corporation.
Windows is a trademark of Microsoft Corporation.
INFOVIEW is a trademark of Unisys Corporation.

using. On the screens illustrated in this guide, input fields are delineated by square brackets []. An input field can extend over several lines to provide sufficient space for you to enter lengthy information. MARC is not case-sensitive; that is, it accepts all input in either uppercase or lowercase letters.

Input fields might be prefilled with default values when displayed. You can either accept the default value as your input, or you can type over the default value with the value you want to enter.

To move the cursor forward to the next input field, use the TAB key or the RETURN key, provided that these keys have not been programmed to execute some other function. On some terminals, the TAB key is labeled with a right-pointing arrow (->) or *SKIP TAB*. To move the cursor back to the previous input field, use the SHIFT-TAB key sequence. On some terminals, you can use a single key labeled with a left-pointing arrow (|<-) and on others you can use the RTAB (reverse tab) key.

Screen Layout

MARC uses various types of screens, including menus, forms, command output screens, and help screens. Certain elements are common to all screens. Figure 2-2 illustrates these elements.

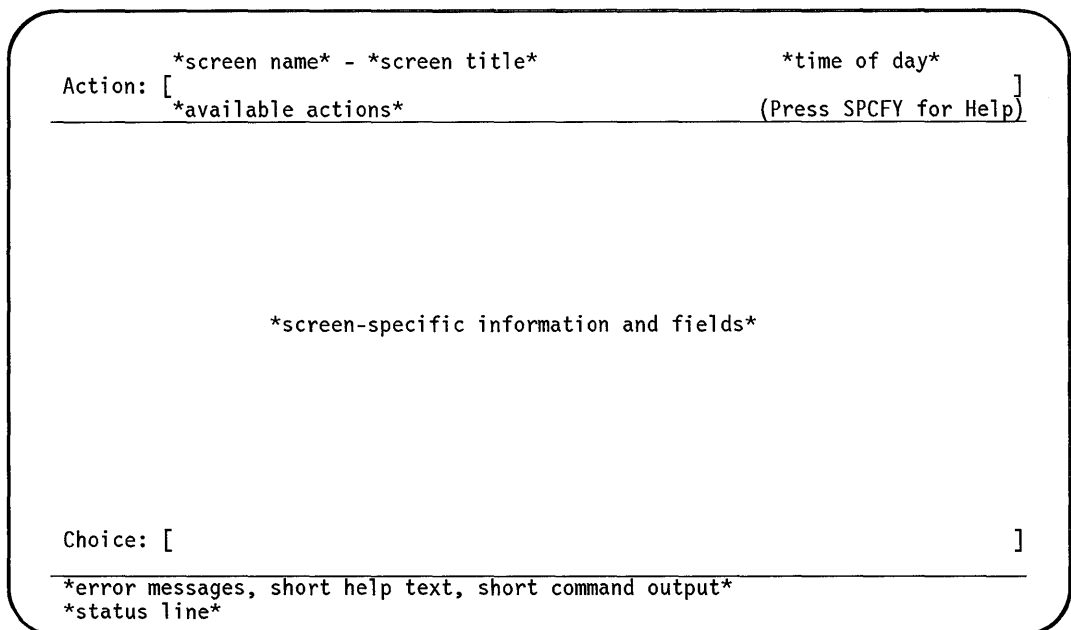


Figure 2-2. Sample Screen Layout

Table 2-1 describes the screen elements illustrated in Figure 2-2.

Table 2-1. Elements Common to MARC Screens

Element	Line Number or Numbers	Explanation
Screen name	1	A unique identifier for a particular screen. Every screen has its own name.
Screen title	1	A more detailed description of the contents of the screen. Every screen has its own title.
Time of day	1	The time of day at which the screen was displayed. Once the screen appears, the time displayed here remains fixed until the next screen is displayed or until the screen is refreshed. For more information, see "Refreshing the Screen," later in this section. The time is displayed in either 12-hour or 24-hour format, depending on the current convention. For more information, see "Choosing a Convention" in Section 9, "Using MARC Internationally."
Action field	2	An input field that appears on every MARC screen except the LOGON screen. You can enter data in this field to perform actions such as displaying another screen or executing a system command.
List of available actions	3	Actions that you can enter in the <i>Action</i> field. Each action causes some other screen to be displayed. The available actions vary from one screen to another. You can abbreviate these actions when you enter them in the <i>Action</i> field. Anything from the first letter to the full spelling of the action is accepted, as long as enough letters are provided to distinguish the action from the other available actions.
"Press SPCFY for Help."	3	A message indicating that you can position your cursor anywhere on the screen, press the SPCFY key, and be presented with help information. See Section 3, "Using the Online Help Facilities," for information about the help feature in MARC.
Screen-specific information and input fields	5-20	On menus, command output screens, and help screens, an area displaying information. On forms, an area containing at least one input field and some information to help you decide what to enter in the input field or fields.
Choice field	21	On menu screens, the field in which you enter the name of a menu selection. Not used on screens other than menus.

continued

Table 2–1. Elements Common to MARC Screens (cont.)

Element	Line Number or Numbers	Explanation
Error messages, short help information, or short command output	23–24	Reserved for the display of error messages or other system messages, for short help information, and for command output that fits on one or two lines. Longer command output is displayed on the OUTPUT screen. See “Displaying Output from Commands” later in this section for information about the OUTPUT screen.
Status line	25	A display of the current dialogue number and hostname. Task messages indicating the various stages of a task are also displayed here.

Refreshing the Screen

You can refresh the MARC screen that is currently displayed on the screen by pressing, in order, the following keys:

1. The *CTRL* (control) key
2. The zero (0) key twice
3. The XMIT key

If your terminal provides a numeric keypad with a double zero (00) key, you can use that key instead of pressing the zero key twice.

Refreshing the screen erases the current display and redisplay the original screen. Also, the status line is refreshed to show only the current window and dialogue number.

Refreshing the screen can be useful in any of the following cases:

- You have used a COMS display command—for example, STATUS—and preceded it with a question mark (?). The output from these commands overwrites part of the MARC screen.
- You enter input in several input fields on a screen and then realize that you want to enter a different set of values. You can always type over the values that you originally entered, but you might prefer to blank out all those values by refreshing the screen.
- You want to update the time of day that appears on the top line of the screen. This information is automatically updated when you refresh the screen.

Transmitting Input

Information you enter at your terminal is sent to MARC only when you press the XMIT key.

Entering input to MARC involves the following steps:

1. Verify that an appropriate screen appears on your terminal, or move to an appropriate screen.
2. Verify that the cursor is properly positioned, or move the cursor to the proper position.
3. Type any information you want to send to MARC in the appropriate area or areas of the screen. In some cases – when choosing a selection key on a menu, for example – the position of the cursor is all the information you need to send.
4. Press the XMIT key.

When this guide provides instructions about entering input, it implies this entire sequence. For example, the instruction “Enter a HELP command” implies verifying that the cursor is located at the beginning of the *Action* field, typing the command word, *and* pressing the XMIT key.

For best results when using MARC, configure your terminal for full-screen transmission. Otherwise, the information transmitted when you press the XMIT key might consist of just the line where the cursor was positioned, or just the data between the home position and the cursor position. See “Configuring Terminals for Full-Screen Transmission” in Appendix B, “Installing MARC on Your System,” for more information.

Action Field Typeahead

MARC enables you to make a sequence of entries in the *Action* field – rather than a single entry – to bypass a series of screens that you do not want to see. This capability is called *typeahead*. MARC executes the entries one at a time in the order you enter them.

When entering a sequence in the *Action* field, note the following guidelines:

- One or more blank spaces must separate the items in the sequence.
- The first item in the sequence must be one of the screen actions listed on line 3 of the current screen. If the action involved is the GO action, follow it immediately by the name of the screen to which the action leads. The GO action might not appear on the screen if the system administrator has restricted its use. In this case, you can use GO only if you have privileged, SYSTEMUSER, or super-user status. See Section 7, “Controlling System Security,” for more information.

- If you use GO followed by a menu name, you can immediately follow this entry with one or more menu selections. The first menu selection must occur on the screen to which the GO action leads. Any further menu selections must each occur on the menu to which the preceding menu selection leads. Menus are described under “Using Menus” later in this section.
- At the end of a GO typeahead sequence for a menu or form, you can enter form field values. These values must apply to fields in the form to which the screen action and menu selections lead. Enter the values for each of the form fields in the order in which the fields appear on the form. You can leave an optional parameter unspecified by entering a question mark (?) in its place. For information on forms, see “Using Forms,” later in this section.

The following guidelines apply to fields entered in a typeahead sequence:

- One or more blank spaces must separate the items in the sequence.
- If you specify fewer fields than appear on the screen, the screen appears with the fields filled in and the cursor positioned at the next field requiring data.
- If you enter data for more fields than there are on the screen, the screen fields are filled in, excess data is written to the *Action* field, and an error message appears.
- If you enter data for the correct number of fields, processing continues as if you had typed directly into the fields on the screen. The next screen displayed after you enter a correct typeahead sequence is the result of processing the screen.
- If you are using the typeahead feature and you need to enter more than one word in a field, enter the field value within quotation marks ("). Otherwise, a blank space indicates that the next word is an entry for the next field. Any typeahead entry within quotation marks is taken as one field value. For example, the following entry invokes the SENDSTA form and fills in the two fields on that form (Station Name and Message):

```
GO SENDSTA TA136 "Please log off"
```

Finding Your Way through the Screens

The first screen that you see after logging on is normally the MARC home menu. If your system administrator has created a customized home menu for your usercode, that menu appears first, rather than the MARC home menu. Custom menus are discussed in Section 8, “Creating Custom Versions of MARC.”

The home menu, labeled *MARC - MENU ASSISTED RESOURCE CONTROL*, provides a starting point for navigating through the MARC menugraph. From this point, the path that you follow through the menugraph depends on the input you enter on each of the screens presented to you.

Eventually, the path you follow through the menugraph reaches a point at which MARC generates a command. MARC builds this command with information you entered during the menugraph traversal. MARC then passes the command to the appropriate portion of the system software, which carries out the command. Commands can be the immediate result of a menu selection, or they can be constructed as the result of

additional information supplied on a form. You can also enter commands directly. For more information on each of these means of generating commands, see “Using Menus” and “Using Forms” later in this section, and Section 6, “Using Commands.”

Using Menus

Using menus in MARC enables you to perform functions without having to learn any command names or command syntax in advance. Menus present you with a list of available selections. Menu selections are sometimes grouped under headings to make it easier to find the task you want to perform. The home menu, which is the first menu in the menugraph, appears automatically once you have logged on. Other menus are reached by way of a series of actions or selections that you enter, starting at the home menu and continuing at each screen that is displayed along the way.

The menu illustrated in Figure 2–3 appears if you choose the JDC selection on the MARC home menu. This menu is typical of the menus MARC displays.

```
JDC - JOB DISPLAY AND CONTROL                                12:34:06
Action:  H0me PReV GO PAREnt COmnd                          (Press SPCFY for Help)
-----
C      Display
J      Completed Jobs and Tasks
MX     Active Tasks in Job Structure
W      Active Jobs/Tasks w/ Displays
Y      Waiting Tasks
TI     Status of a Task
CU     Time Accumulator for Task
OT     Core Usage for Task
CO     Contents of Stack Cell
MSG    Compiling Sequence Number
SMSG   Session Messages
FIRST  System Messages
SHOW  First Job in Each Queue
      All Jobs in One Queue
      Control Operations
      DS Discontinue Task
      ST Temporarily Suspend Task
      OK Resume Suspended Task
      HI Cause EXCEPTIONEVENT of Task
      AX Pass Text to Program
      FA Change a File Attribute
      OF Continue w/o Optional File
      RM Remove DUP LIBRARY Disk File
      FR Tape Read is Final Reel
      TIME Assign Job Start Time and Date

Choice: _____
```

Figure 2–3. Sample MARC Menu

For information about the screen actions listed on line 3, see “Using the Action Field” later in this section.

Menu Selections

The middle portion of the screen lists the menu selections. The sets of selections presented differ from one menu to another. The selections are presented in one, two, or three columns. The menu in Figure 2–3 is an example of the two-column format.

For each selection, the menu displays a selection key, followed by a selection description:

- A *selection key* is a group of up to six alphanumeric characters that represent the selection.
- A *selection description* presents a brief summary of what the selection does.

In the example of the JDC menu presented in Figure 2–3, the first selection has a selection key of *C*, followed by *Completed Jobs and Tasks* as its description.

Using a Mouse to Make Menu Selections

If you have a Micro A system, you can use the system mouse to make menu selections when running MARC at the ODT. When a menu appears, you can make a menu selection by moving the cursor to the appropriate selection and double-clicking the left mouse button. (Clicking the left mouse button once positions the cursor only.) The cursor can be on the selection key or the description following the name.

You can use the same procedure to select one of the options displayed for the *Action* field at the top of the menu or form. You can request help information for any screen element by moving the cursor to the appropriate field and clicking the right mouse button. For information on the Micro A system mouse, see the *A Series Micro A System Installation Guide*.

On any A Series system, if your terminal is a Unisys personal computer (PC) or a compatible PC running the PCMARC user interface, you can use the PC mouse to make menu selections. The elements on PCMARC screens correspond exactly to those on standard MARC screens, but the screens are laid out differently. The technique for making PCMARC menu selections is described in the *A Series PCMARC Installation and Operations Guide*.

Using Forms

Forms are information-gathering screens. MARC uses each screen to construct a specific command internally. Typically, a form presents several input fields along with prompts for the values that the fields are to hold. Completing the form and then transmitting causes the command to be carried out.

You can display a form by choosing a menu selection or by using one of the actions listed on line 3 of any menu or form. Figure 2–4 presents an example of a form.

```

          COPY - Copy Files Or Directories                               15:45:38
Action: [ H0me PRev GO PARENT COmnd TAsk                               ]
          (Press SPCFY for Help)
File and/or Directory List . . [ ]

From: Device Type: 1:DISK 2:TAPE 3:CDROM . [DISK ]
      Familyname (or Tapename) [ ] (Familyindex). . . [ ]
      (Hostname). . . . . [ ] (Serial Number). [ ]

To: Device Type: 1:DISK 2:TAPE . . . . . [DISK]
     Familyname (or Tapename) [ ] (Familyindex). . . [ ]
     (Hostname). . . . . [ ] (Serial Number). [ ]
Remove file(s) with duplicate names? (1:YES 2:NO). . . . . [YES]
Compare copied file(s) to original file(s)? (1:YES 2:NO) . . . . . [NO]

----- System will select default values for the following, if left blank -----

File transfer service to be used. (1:HostServices 2:NFT). . . . . [ ]
Recover an earlier aborted file transfer. (1:YES 2:NO). . . . . [YES]

```

Figure 2-4. Sample MARC Form

For information on the screen actions listed on line 3, see “Using the Action Field” later in this section.

Form Input Fields

The middle portion of a MARC screen contains the form input fields. These fields enable you to enter information to the system, but they might contain default values when the screen is displayed. You can type over default values if you prefer other values. In the example of the COPY form shown in Figure 2-4, the system has supplied *DISK* as the default value for the device type.

Each form field has descriptive text beside it that requests the information that you are to enter in that field. This descriptive text is called a *prompt*. If the entire prompt for a field is enclosed in parentheses, that field is optional; you can choose whether to fill it in. In the example of the COPY form presented in Figure 2-4, the *(Familyindex)*, *(Serial Number)*, and *(Hostname)* fields are optional. Normally, you must complete other form fields, though in a few cases a field might become optional owing to input values entered in other fields. In the COPY example shown in Figure 2-4, the *Familyname (or Tapename)* field requires input if the device type specified is *DISK*. This field is optional, however, if the device type specified is *TAPE*.

In some cases, the prompt for an input field includes a list of the possible values that you can enter in that field. In Figure 2-4, for example, *DISK*, *TAPE*, and *CDROM* are listed as the possible values you can enter in the *Device Type* field. When these possibilities are preceded by a number and a colon (:), you can simply enter the number

in the input field if you like. For example, instead of entering *DISK* in the *Device Type* field of the *COPY* form, you can enter the number *1*.

Command Generation

When you transmit a form, MARC uses the values supplied in the form fields to generate a command internally. The command is automatically executed, and if the command output is one or two lines, it is displayed at the bottom of the screen. However, a few of the forms available in MARC initiate tasks, which are described under “Running a Task” in Section 4, “Running Programs from MARC.”

Using the Action Field

You use the *Action* field to enter actions that you want to execute. Certain actions are available for use on most screens, while others are available only when you are listing the contents of a file.

Actions Available on Most Screens

The following paragraphs describe the actions you can enter in the *Action* field of most menus and forms. Not every action described is available on every screen. Actions available on a given screen appear immediately below the *Action* field on that screen. See “Running a Task” in Section 4, “Running Programs from MARC,” for the actions available for running a task. For more information on the actions available for obtaining help information online, see Section 3, “Using the Online Help Facilities.”

HOMe

This action displays the home menu, the menu that is presented just after you log on. The home menu can be different for each usercode. See “Setting the Usercode Default Home Menu” in Section 8, “Creating Custom Versions of MARC,” for information on how the home menu is determined.

PRev

This action displays the screen that was last displayed. Often this is the same screen you reach by using the PARENT action, but not always. MARC stores the last 15 screens that were displayed, and by using the PRev action repeatedly, you can redisplay these screens in reverse order.

You can also use the PRev action to redisplay help information from the previous help request. MARC stores the last 10 help screens that have been displayed, and you can redisplay them in reverse order by transmitting successive PRev actions. While you are displaying help screens, the PRev action applies only to help screens. You must use the RRETURN action to display previously displayed screens and forms.

Starting to Use MARC

For example, suppose you are viewing screens of help text for the *Action* field. To page backward through the text for the current help request, you use the minus (-) scrolling action. However, to display the help information from the previous help request, you use the PRev action. If at this point you want to return to the form or menu from which you made the current help request, you must use the RReturn action rather than the PRev action.

GO

Typing *GO* followed by a screen name causes the specified screen to be displayed. The screen name is the short name that is displayed in the upper left corner of each menu and form. The *GO* action is useful when you already know the name of the screen that you want to reach. *GO* requires the entire name of the screen; it rejects abbreviations.

You can also use the *GO* action to display a screen in an alternate menugraph. For more information on this capability, see “Declaring a Substitute Menugraph” and “Bypassing the Menugraph Search Order” in Section 8, “Creating Custom Versions of MARC.”

The system administrator can restrict this action by using the Interactive Menugraph Generator (IMG) to remove *GO* from the action line of the system menugraph. Use of the *GO* action is then limited to privileged users, users with SYSTEMUSER usercodes, and super users – users who log on under the asterisk (*) usercode from a station that has been defined as a super-user-capable station. Any attempt to enter *GO* under any other circumstances results in an error message. See Section 7, “Controlling System Security,” for more information on MARC security features.

PArent

This action displays the menu containing the selection that led to the current screen. This parent menu also contains selections for related screens. For example, if you find yourself at a screen that seems close to the one you want, displaying the parent menu might lead you to the correct screen.

COmnd

This action initiates the use of commands. Because you can enter commands that fit on a single line directly in the *Action* field, the *COMND* action is optional, with one exception: the system command C (Completed Mix Entries) must be preceded by the *COMND* action.

For commands longer than the *Action* field, the *COmnd* action displays a full screen, named *COMND*, for command input. See “Entering Commands” in Section 6, “Using Commands,” for more information on the *COMND* action and the *COMND* screen.

KEys

This action is available from a screen of long help text and displays all the help keywords that are available through MARC. You can also request this information by

entering *HELP KEYS* in the *Action* field of most MARC screens. For more information, see “Using the HELP Command” in Section 3, “Using the Online Help Facilities.”

REturn

If you are using commands, this action causes the screen that generated the command to be redisplayed on the terminal. If you are displaying a help screen, this action causes MARC to return to the screen from which you requested help. In either case, the screen displayed could be a menu, a form, or the COMND screen.

To run a task from MARC, you can use the REturn action on the TASKVIEW screen to return to the TASKSTATUS screen. When used on the TASKSTATUS screen, the REturn action displays the screen from which you initiated tasking.

Plus (+)

This action is used for scrolling forward through command output, help text, or the list of active tasks. Transmitting a plus sign (+) in the *Action* field scrolls the display one page forward. Transmitting a plus sign (+) followed by a number causes the command output to be scrolled forward the specified number of lines. Transmitting a plus sign (+) followed by a number and the letter *p* causes the command output to be scrolled forward the specified number of pages. For example, transmitting *+3p* scrolls the command output forward three pages.

If there is more command output available, MARC displays a plus sign (+) in the *Action* field. If you have reached the end of the command output, MARC displays *REturn* in the *Action* field. These displays are default values that you can transmit or type over.

Minus (-)

This action has the same capabilities as the plus (+) action except that it scrolls the output backward rather than forward. After you have used the minus (-) action, MARC displays a minus sign (-) in the *Action* field if there is more output to be reached in that direction. If you have scrolled all the way back to the beginning of the command output, MARC displays a plus sign (+) in the *Action* field.

Actions Available Only with File Lists

The following actions are available only on the output screen displayed when you use the LIST command or selection to list the contents of a file.

First

This action displays the first page of the file you are listing.

LAsT

This action displays the last page of the file you are listing.

SEQ

This action displays the file beginning with a specific sequence number. The only valid entry is *SEQ* followed by a line number. If you are listing a file from a LIST screen or command where you set the RELATIVE option, SEQ uses the relative line number to access your file. A file using relative line numbers is numbered sequentially by increments of 1, with the first line of your file displayed as line number 1. For example, if you enter *SEQ 500* and the RELATIVE option is set, the 500th line of the file is displayed regardless of its actual sequence number.

COL

This action shifts the column range for viewing the current file. You can enter the COL action in one of the following formats:

- *COL <start column>*
- *COL <start column> - <end column>*
- *COL <start column> - END*

Using the Choice Field

The *Choice* field appears at the bottom of menus. You use this field primarily to enter the selection key for the menu selection you choose. The selection key and the name of the resulting screen are often the same, but not always. The selection key can be up to six letters long, but you need enter only as many letters as are necessary to differentiate it from the other selection keys on the current menu. On the JDC menu illustrated in Figure 2-3, for example, entering *R* is sufficient to identify the RM selection because no other selection begins with that letter.

You can also use the *Choice* field to enter parameters associated with a given selection key. By doing so, you can bypass the presentation of a form that asks for the values to use for these parameters. The parameter values must be listed after the selection key in the same order in which they would appear on the form. For example, you can enter *TOOLS IMG* in the *Choice* field of the MARC home menu to run IMG.

You can also use this field to enter a HELP command. See “Using the HELP Command” in Section 3, “Using the Online Help Facilities,” for more information.

Choice Field Typeahead

The *Choice* field provides a typeahead capability that is similar to the typeahead capability provided by the *Action* field. The only difference is that you cannot include screen actions in the *Choice* field. You can include only menu selection keys or form

parameters in *Choice* field typeahead. See "Action Field Typeahead" earlier in this section for more information.

Command Generation

Transmitting a selection key can cause another menu or a form to be presented, or it can cause a command to be generated. If the command output is one or two lines, the output appears at the bottom of the current screen. Otherwise, a command output screen containing the output appears. If you transmit a selection key for a selection that requires a parameter and you omit the parameter value, a form requests the value of that parameter.

Displaying Output from Commands

When MARC has executed a command and the command output is longer than two lines, the OUTPUT command output screen is displayed to show the system response to the command. Commands are generated directly by some menu selections and by most forms. You can also initiate commands by entering them directly in the *Action* field of most screens or on the COMND command input screen, which is described in Section 6, "Using Commands."

Figure 2-5 presents an example of the OUTPUT command output screen.

OUTPUT - MARC COMMAND OUTPUT				04:11 PM		
Action: [REturn]		
H0me GO REturn COMnd + -				(Press SPCFY for Help)		
Response returned at 04:11 PM:						
Currently Printing Requests						
Job	Request	Amt Left	Done	(User)	Job Name	Device
7155	20661	385 Ln	9%	(ACCOUNTS)	"Session"	LP5
8487	20366	4000 Ln		(SYSLOG)	OBJECT/TEST	LP2032A
5796	20586	3965 Ln		(INVENTORY)	"Session"	IP7
5308	20569	446 Ln		(SALES)	"Session"	LP4
5235	20570	406 Ln	0%	(PROJECTS)	"Session"	LP6
PS SHOWREQUEST P						

Figure 2-5. Sample OUTPUT Screen

The bottom two lines of the screen display the command that was executed. If the command is too long to fit on the two lines, a truncated version of it is displayed. The middle portion of the screen contains the output from the command.

Continued Information

Command output can be of any length, so it might be longer than can fit on one command output screen. You can use the plus (+) and minus (-) actions to scroll through multiple screens of command output. When you scroll the command output forward, the title of the screen changes from *OUTPUT - MARC COMMAND OUTPUT* to *CONTINUATION - MARC COMMAND OUTPUT*. The message displaying the time the response was returned disappears, and the command output display continues.

End of Output

You cannot scroll past the end of command output. For example, if you enter a plus sign (+) to scroll the output forward one page and only five lines of output remain for display, the output is scrolled just five lines forward. In this case, a plus sign (+) appears in the left column of the screen next to the first new line of information that is presented.

The default screen action displayed in the *Action* field changes from a plus sign (+) to *REturn* when you have reached the end of the command output. However, you can always use the minus (-) action to scroll the command output backward if you want to review information earlier in the display.

Logging Off

You can end a MARC session with the commands *BYE*, *HELLO*, or *SPLIT*, or with the *COMS CLOSE* command.

Using *BYE*

You can use the *BYE* command only at the MARC/1 dialogue. The *BYE* command closes all the MARC dialogues, as well as all the dialogues on the Command and Edit (*CANDE*) window and all other open windows.

The *BYE* command is rejected if any tasks initiated during the MARC session are still running under any MARC dialogue. You must either wait for the tasks to finish running or terminate them before logging off. However, *BYE* ends the MARC session even if there are tasks running under the *CANDE* window or other message control system (*MCS*) windows. As a result, these tasks are all terminated immediately.

The *BYE* command is also rejected if any of the window dialogues have *tanked output* – that is, program output that has been temporarily stored at the terminal. This output remains stored until you display it. For example, if you run a program in window *CANDE/1* but switch to MARC/1 and enter *BYE* before the program finishes displaying its output, the system displays the following message:

```
Window CANDE/1 cannot be closed at this time. Tanked outputs exist.  
Try using the RESUME command or the BYE command with the PURGE option.
```

To suppress the stored output, enter *BYE PURGE* in the *Action* field. To see the output, enter *RESUME CANDE/1*. The output overwrites the MARC/1 window. When the output display is complete, press the HOME key, and then enter *BYE*.

When the BYE command is accepted, any printer output that was generated under any of the MARC dialogues is sent to the print queue immediately.

Using HELLO

You can use the HELLO command to end the current session. The command has the same effects and the same restrictions as the BYE command, with two exceptions:

- You can use the HELLO command only in the *Action* field at the MARC/1 dialogue.
- After ending the current session, the HELLO command logs you on under a specified usercode.

The HELLO command has the following format:

```
HELLO <usercode> <password>
```

To start the new session, enter the usercode under which you want to log on and the password associated with that usercode.

On some systems, security restrictions prevent you from using the full form of this command. If this is the case on your system, enter just *HELLO* in the *Action* field to display the LOGON screen and follow the normal log-on procedure.

Using SPLIT

You can use the SPLIT command at any dialogue. SPLIT ends the current session at that dialogue and immediately initiates a new session under the same usercode. Any printer output generated at the current dialogue is queued for printing.

Using ?CLOSE

You can use the following COMS command to close specific MARC dialogues:

```
?CLOSE /<dialogue number>
```

This command ends the session in the specified dialogue and discontinues any tasks running in that dialogue. You can use this command to end the session in any MARC dialogue except MARC/1.

Section 3

Using the Online Help Facilities

One of the major advantages of MARC is that it provides comprehensive online help text for every task you can perform through MARC. The help text is *context-sensitive*, meaning that it is specifically related to the task you are currently performing. Each menu has help text for the menu itself and every one of its selections; each form has help text for the form itself and every one of its fields. MARC also provides online definitions of terms used in the help text. In addition, MARC provides a number of online tutorials about MARC and about A Series systems in general.

For each menu, menu selection, form, and form field, two levels of help text are available. *Short help text* is a brief explanation (one or two lines) of the topic in question. If you need more information, *long help text* provides expanded coverage of the topic, often extending over several display screens.

Short Help Text

To request short help text related to any menu selection or form field, move the cursor on the terminal screen to the form field or menu selection in question, and then press the SPCFY key. When you position the cursor for short help on a menu selection, you can place it either on the selection key or on the text associated with it. When you position the cursor for short help on a form field, you can place it anywhere within the field: either on the prompt text or within the input area enclosed by delimiters.

To display short help information about a menu or form as a whole, place the cursor on the screen title and press the SPCFY key. To display short help information about the list of available actions, place the cursor anywhere between the *Action* field delimiters and press the SPCFY key.

The short help text for each menu selection and form field is limited to two lines and is displayed on the two bottom lines of the screen. The message *Press SPCFY for Help* at the end of line 3 changes to *SPCFY again for more*.

For example, assume that you have selected the JDC menu (or the JC menu, depending on your security privileges) from the MARC home menu, and you want to know what the AX selection does. Move the cursor to the portion of the menu occupied by the AX selection. You can place the cursor either on the selection name itself, or on the *Pass Text to Program* description. Press the SPCFY key. The short help text for the AX selection appears at the bottom of the screen, as illustrated in Figure 3-1.


```
JDC - JOB DISPLAY AND CONTROL                                19:11:36
Action: [ Home PRev GO PArnt COmnd                          (SPCFY again for more) ]

      Display
C     Completed Jobs and Tasks          DS   Control Operations
J     Active Tasks in Job Structure     ST   Discontinue Task
MX    Active Jobs/Tasks w/ Displays     OK   Temporarily Suspend Task
W     Waiting Tasks                    HI   Resume Suspended Task
Y     Status of a Task                  AX   Cause EXCEPTIONEVENT of Task
TI    Time Accumulator for Task         FA   Pass Text to Program
CU    Core Usage for Task               OF   Change a File Attribute
OT    Contents of Stack Cell            RM   Continue w/o Optional File
CO    Compiling Sequence Number         FR   Remove DUP LIBRARY Disk File
MSG   Session Messages                 TIME  Tape Read is Final Reel
SMSG  System Messages
FIRST First Job in Each Queue
SHOW  All Jobs in One Queue

Choice: [ ]

This choice sends text to tasks identified by a list of mix numbers,
when those tasks need information to continue processing.
```

Figure 3-1. Sample Short Help Text Display

If you leave the cursor where it is and press the SPCFY key again, MARC displays the long help text, providing further information on the same subject.

To bypass the long help information, press the HOME key to move the cursor to the *Action* field, from where you can move to another MARC screen or enter a command. Or, you can press the TAB key to move the cursor to the *Choice* field, from where you can select AX or make another selection on the JDC menu.

Long Help Text

Continuing the example just discussed, if you press the SPCFY key a second time for the AX selection, MARC displays the more detailed long help information on your screen. Figure 3-2 illustrates the first page of the long help text for the AX selection.

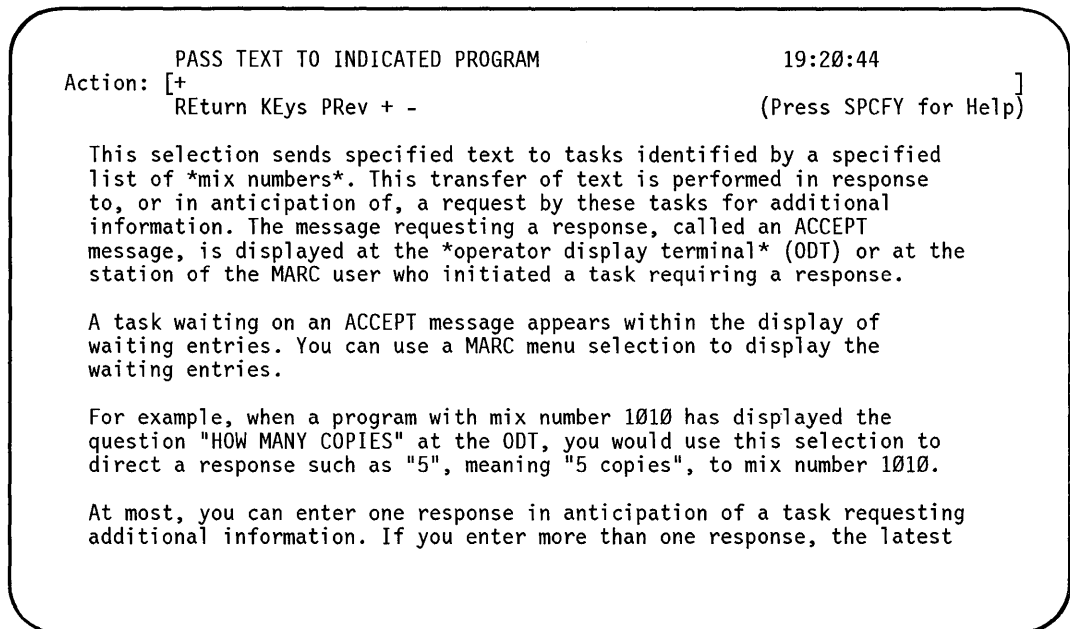


Figure 3-2. Sample Long Help Text Display

The body of the screen displays the help text. Long help text can be of any length. If the text extends beyond one screen, as in this example, you can use the plus (+) or minus (-) actions to scroll forward or backward through the text. When you have finished reading the text, enter *R* (for Return) in the *Action* field to return to the JC menu.

For information on all the screen actions listed on line 3, see “Using the Action Field” in Section 2, “Starting to Use MARC.”

Online Definitions of Terms

Some of the terms that appear in help text are enclosed in asterisks (*). These terms have definitions available online. For example, the help text in Figure 3-2 indicates that the term *mix numbers* has an online definition available. To display the definition associated with the term, place the cursor anywhere between the two asterisks, and press the SPCFY key. The glossary definition is displayed on another help screen, overwriting the first help screen.

Figure 3-3 illustrates the screen displayed when the user selects the term **mix numbers** on the screen illustrated in Figure 3-2.

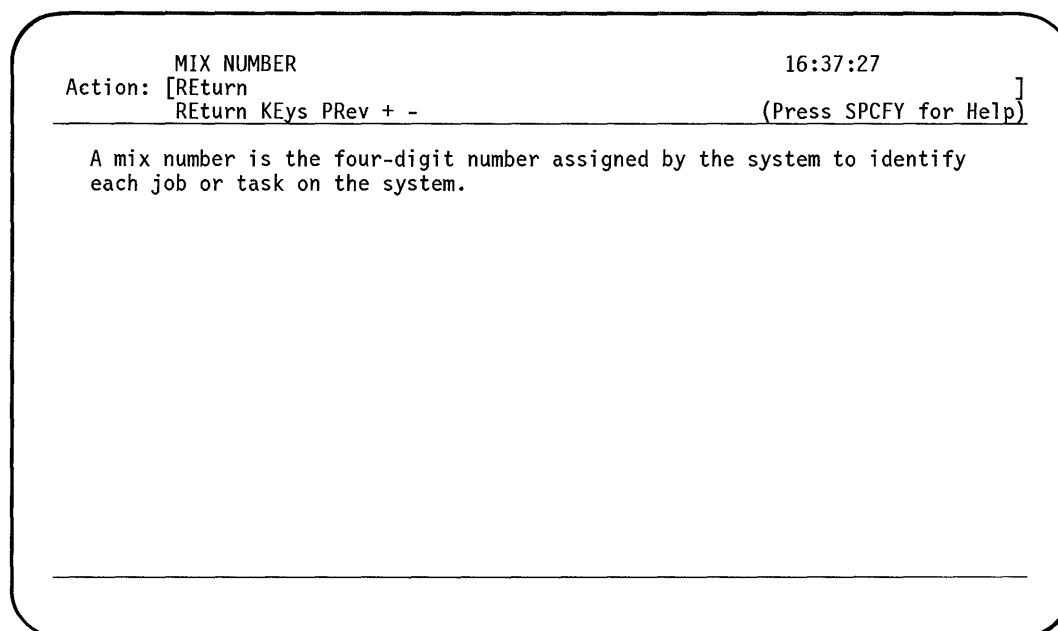


Figure 3–3. Sample Online Glossary Definition

When you have finished reading the online definition, you can enter one of the following characters in the *Action* field:

- *P* (for Previous) to redisplay the long help text that was overwritten by the definition—for the AX selection in the example.
- *R* (for Return) to return to the menu from which help text was requested—the JC menu in the example.

Not all the terms that have online glossary definitions are enclosed in asterisks in help text. Asterisks are used sparingly to maintain the readability of the help displays. If you require a definition for a term that is not enclosed in asterisks, you have several options:

- Move the cursor to the term and press the SPCFY key. If help information is available for that term, that text appears. Otherwise, MARC displays a message stating that no help text is available for that term.
- Enter *HELP <term>* in the *Action* field. You can use this method for a word or a phrase, even if the term does not appear on the help screen. If the term consists of more than one word, enclose the term in asterisks.
- Enter *HELP MARC TERMS* in the *Action* field, or choose the *TERMS* selection on the *TEACH* menu. Either method displays an alphabetic list of those terms for which an online definition exists in the MARC help text.
- Choose the *GLOSS* selection on the *TEACH* menu. This selection provides access to the online A Series Master Glossary, which contains many more definitions than are available in the MARC help text.

For more information on the last three of these options, see “Using the HELP Command” and “Using the TEACH Menu” later in this section.

Using the HELP Command

The HELP command enables you to display any section of MARC help text, as well as text from any other help book in use on your system. You can enter this command in either the *Action* field or the *Choice* field of any MARC screen that accepts commands. The HELP command has the following syntax:

```

┌─ HELP ─┐ ┌─<selection key>──────────────────────────────────────────┐
└─ TEACH ┘ └─<keyword>──────────────────────────────────────────────────┘
                    └─ IN ─<book>──────────────────────────────────────────┘
  
```

This command displays the help information associated with the specified selection key or keyword. (In this command, *TEACH* is a synonym for *HELP*.) A *selection key* is the group of up to six letters used to identify a particular selection on a menu. A *keyword* identifies a section of help text related to a particular subject.

A MARC menugraph always includes a default book for help information. If you enter *HELP* on any screen other than the TEACH screen and omit the *IN <book>* specification, MARC searches for the specified help text in the default book. If you are on a help screen and enter *HELP* without the *IN <book>* specification, MARC searches for the specified text in the currently displayed book.

How to Enter HELP

You can use the HELP command with a keyword from any screen. You can use the HELP command with a selection key only from a menu, not from a form or a help screen.

You can enter the selection key or the keyword in either uppercase or lowercase characters, or in a combination of both. You can also abbreviate a selection key or a keyword if you supply enough letters to distinguish it from all the other available keys and keywords.

If you enter the HELP command on a menu, MARC assumes that the word following *HELP* is a menu selection key, unless you enter a word or phrase enclosed in asterisks (*). If you enter a single word or phrase without asterisks and there is a matching selection key displayed on the menu, MARC displays the long help information for that selection. The effect of the command is the same as if you had placed the cursor on the selection and pressed the SPCFY key twice. See “Short Help Text” and “Long Help Text” earlier in this section for a discussion of how you can obtain help information by positioning the cursor on the item in question and pressing the SPCFY key.

If there is no matching selection key displayed on the menu, or if you enter the command on a screen other than a menu, MARC assumes that the word entered is a help text keyword and displays help information associated with that keyword.

Finding Keywords

To determine what keywords are available, enter *HELP KEYS* in the *Action* or *Choice* field. *KEYS* is a keyword reserved especially for this purpose. The *HELP KEYS* command displays an alphabetic list of all the available keywords.

For MARC help text, this list is extremely long. You can use the plus (+) and minus (-) actions to scroll through the list. For more information about these actions, see “Actions Available on Most Screens” in Section 2, “Starting to Use MARC.”

Many keywords are words or phrases that are suggestive of the subject described in the associated help text. The *KEYS* command is equivalent to the *KEys* action that is available on help screens, except that the *HELP KEYS* command is available on all MARC screens that accept commands.

While the list of keywords is displayed, place the cursor on any keyword in the list and press the *SPCFY* key to display the help text associated with that keyword.

Finding Reserved Keywords

REFERENCES and *TOPICS* are additional reserved keywords used by the A Series Help Utility in every help book it creates. You can enter *HELP* or *TEACH* in the *Action* field followed by one of these reserved keywords. The reserved keywords and their functions are as follows:

- **KEYS**
Displays a list of all keywords in the help book.
- **REFERENCES**
Displays a list of all references defined in a help book, as described in the *A Series Help Utility Operations Guide*. A reference is a name used in help text to refer to another help book. Place the cursor on any help book reference and press the *SPCFY* key to display the *TOPICS* of the associated help book.
- **TOPICS**
Displays a list of all the section titles in the help book.

Selecting Other Help Books

You can use the *book* specification with the *HELP* command to display help information from help books other than MARC. Your installation might provide several help books on a variety of topics. Your system administrator can tell you which help books are available on your system.

Using the TEACH Menu

The first selection on the MARC home menu is the *TEACH* selection, which provides access to an extensive set of online tutorial information about MARC and about

A Series systems in general. Figure 3-4 illustrates the menu that appears when you choose the TEACH selection.

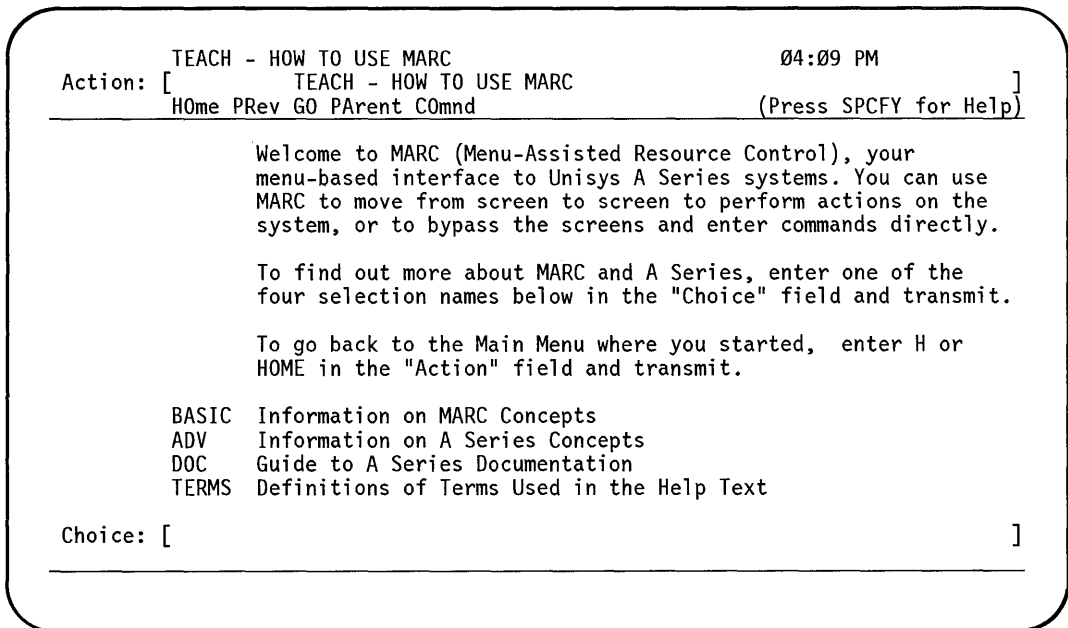


Figure 3-4. TEACH Menu

Table 3-1 lists the TEACH menu selections and the information they provide.

Selection	Information Provided
BASIC	Information on using MARC, including how to use the screens, how to navigate through the menus, and how to display help text.
ADV	Advanced information on using A Series systems in general, including details on the use of COMS windows, file management, program development, and printers.
DOC	The online <i>A Series Documentation Library Overview</i> , a guide to the entire suite of A Series printed documentation. You can access information about an A Series manual in several ways: by function, by product, or alphabetically by manual title.
TERMS	An online glossary containing definitions of terms used in the MARC help text.
GLOSS	The online A Series Master Glossary, providing definitions of an extensive range of terms used in connection with A Series systems. Full instructions are provided on the screen.

If you are new to MARC or to A Series systems, the BASIC and ADV tutorials provide a useful way to familiarize yourself with the most important aspects of the system.

For a more comprehensive introduction to A Series systems, see the *A Series Systems Functional Overview*.

Using a Mouse to Display Help Text

If you have a Micro A system, you can use the system mouse to display help text when running MARC at the ODT. How you use the mouse depends on whether you want to display help text for a menu or for a form.

- On a menu, you can get help for a menu selection by moving the cursor to that selection and clicking the right mouse button. The cursor can be on the selection name or the description following the name. This procedure is equivalent to moving the cursor with the arrow keys and pressing the SPCFY key.
- On a form, you can get help for a field by moving the cursor between the field delimiters and clicking either the left or right mouse button.

For information on the Micro A system mouse, see the *A Series Micro A System Installation Guide*.

If your terminal is a Unisys PC or a compatible PC that supports the PCMARC user interface, you can use the PC mouse to display help text. The elements on PCMARC screens correspond exactly to those on standard MARC screens, but the screens are laid out differently. The technique for displaying help text with PCMARC is described in the *A Series PCMARC Installation and Operations Guide*.

Section 4

Running Programs from MARC

You can run any system or user-written program from MARC, provided that your security level permits. The MARC home menu provides the following two selections that enable you to run programs:

- The START selection initiates a program from a Work Flow Language (WFL) job.
- The RUN selection initiates a program directly, as a task.

The *Work Flow Language* (WFL) is the Unisys language in which the statements are written that define exactly what the job does.

Both tasks and jobs initiated by MARC inherit, by default, various attributes from the current MARC session. These task and job attributes include:

- JOBNUMBER (which is set to the SESSION number)
- STATION
- SOURCESTATION
- USERCODE
- CHARGECODE
- FAMILY
- LANGUAGE
- JOBSUMMARY
- JOBSUMMARYTITLE
- PRINTDEFAULTS
- BACKUPFAMILY

Distinguishing between a Task and a Job

A *task* is typically a run of some program that performs a data processing function, such as a payroll program to print paychecks, or an accounting program to produce financial reports.

A *job* is a batch process whose only purpose is to initiate a program. A job can also initiate multiple programs in some particular order, perhaps also monitoring the progress of these programs and helping to synchronize their activities.

For example, a job might initiate the compilation of a particular program, initiate the execution of the resulting program code file, and initiate the printing of output generated by the program. The system handles the compilation, execution, and printing as three separate tasks, all initiated by the one job.

Running Programs from MARC

You can initiate a task from a menu by using the ADD, COPY, UTIL, RUN, or START menu selection, or any of the system utility menu selections. You can initiate a task from a menu by entering a WFL *COPY*, *RUN*, or *START* statement as a command, or from a WFL program.

All these menu selections and commands initiate synchronous tasks. The START menu selection and START command first initiate the WFL compiler as a synchronous task. Then, if compilation is successful, the task causes an asynchronous job to be initiated before going to completion. The terms *synchronous task* and *asynchronous job* are explained in the following paragraphs.

Synchronous Tasks

A synchronous task is a task that runs under the direct control of the agent that initiated it. In MARC, you control a synchronous task (from here on called simply a *task*) by entering actions on the TASKSTATUS screen or by interacting with the task through a remote file. Only one task at a time can be running on a MARC dialogue. However, you can use the ON command to switch to another dialogue while the task is running in order to start another task. See Section 5, “Using COMS Windows and Dialogues,” for information on MARC dialogues. MARC runs the following types of code files as synchronous tasks:

- Code files without parameters
- Code files with array parameters
- Code files with real parameters

Asynchronous Jobs

An asynchronous job is a job that runs independently of the agent that initiated it. In MARC, your only means of controlling an asynchronous job (from here on called simply a *job*) is through system commands. Multiple jobs can be started from a MARC dialogue. You can also run a task while one or more jobs are running.

Starting a WFL Job

A WFL job is a series of WFL statements contained in a file of type JOB. Your site might have predefined WFL jobs for your use, and you can write your own. See “WFL Statements Entered as Commands” in Appendix C, “Commands Available in MARC,” for a list of the WFL statements. For detailed descriptions, see the *A Series Work Flow Language (WFL) Programming Reference Manual*.

You can start a WFL job in MARC by means of the START screen or by entering a command in the *Action* field.

Starting a WFL Job from the START Screen

To start a WFL job from the START screen, perform the following steps:

1. Do either of the following:
 - Choose the *START* selection on the MARC home menu.
 - Enter *GO START* in the *Action* field of a MARC screen.

In either case, the *START* screen appears, as illustrated in Figure 4-1.

```

START - Starting Work Flow Language Jobs          17:02:31
Action: [                                         ]
        H0me PRev GO PAREnt COmnd                (Press SPCFY for Help)
-----
Filename of job file: [                           ]
Parameter(s): . . . [                             ]
                                           ]

Optional Starting TIME and DATE:
ABSOLUTE time and date . . . . . [ ] [ ] [ ]
      or
RELATIVE to current time and date:
  In how many hours and minutes from the current time? ( ? 24 hrs ) [ ]
  In how many days from today? . . . . . ( 0-99 days) [ ]
    
```

Figure 4-1. *START* Screen

2. Enter the name of the file containing the WFL job statements in the first field, *Filename of job file*, and any parameters the job requires in the second field, *Parameter(s)*. For example, suppose you have a job that in CANDE requires the following command:

```
START COMPILE/JOB("SOURCE=PROG1")
```

To start this job in MARC, you can enter the information on the *START* screen. Figure 4-2 illustrates a filled *START* screen.

```

          START - Starting Work Flow Language Jobs          17:06:58
Action: [
      H0me PRev GO PArEnt COmnd          (Press SPCFY for Help) ]
-----
Filename of job file: [COMPILE/JOB
Parameter(s): . . . [SOURCE=PROG1
]
]
Optional Starting TIME and DATE:
          hh:mm    AM or PM          mm/dd/yy
ABSOLUTE time and date . . . . . [ ] [ ] [ ]
or
RELATIVE to current time and date:          hh:mm
  In how many hours and minutes from the current time? ( ? 24 hrs ) [ ]
  In how many days from today? . . . . . ( 0-99 days ) [ ]
-----

```

Figure 4–2. START Screen (Filled)

In the *Parameters* field, you can omit the parentheses or quotation marks (") surrounding the source file equation that would be required with CANDE.

Use the remaining fields on the START screen only if you want to delay the start of the job until a certain time.

3. When you have filled in all the necessary fields, press the XMIT key.

When you press the XMIT key, the job starts immediately, if possible; otherwise, it is placed in a job queue, in which it waits its turn to begin. When the job starts, it initiates the first task and the following message appears on the status line:

Task initiated.

For the procedure after this point, see “Displaying the TASKSTATUS Screen” later in this section.

Starting a WFL Job from the Action Field

You can also start a WFL job from the *Action* field of any MARC screen simply by entering the START command just as you would in CANDE. If the START command is too long for the field, enter CO and transmit. Doing so displays a full screen in which you can enter the START command.

Running a Task

You can run a user-created or system program or utility as a task through MARC. Tasks are initiated implicitly when you make certain menu selections, such as the system utility selections on the UTIL menu, or when you complete certain forms, such as the COPY form to copy a file.

You can run a task in MARC by means of the RUN screen, or by entering a command in the *Action* field.

Running a Task from the RUN Screen

To run a task from the RUN screen, do either of the following and press the XMIT key:

- Choose the RUN selection from the MARC home menu.
- Enter *GO RUN* in the *Action* field of a MARC screen.

Either method results in the display of a series of up to three screens for gathering the information needed to initiate the task, as follows:

- The first screen (the RUN screen) prompts you for the name of the code file to be executed and for any parameter or task values.
- The second screen (the FILEEQUATE screen) enables you to establish file equations and attributes for one or more files.
- The third screen (the TASKATTR screen) enables you to assign task attributes for the task to be initiated.

The following text describes these screens.

Defining the Task to Be Run

On the first screen, RUN, you enter the name of the code file the task is to execute. Figure 4-3 illustrates this screen.

Running Programs from MARC

Action:	RUN - EXECUTION OF A TASK	04:40 PM
	[Home PRev GO PArEnt COmnd	(Press SPCFY for Help)
Task name	[]
Parameter	[]
Value	[]
	[]	The parameter should be treated as alphanumeric.
	[]	File equations are needed.
	[]	Task attributes are needed.

Figure 4-3. RUN Screen

Enter the following information into the RUN screen, and then press the XMIT key:

1. Enter the name of the code file in the *Task name* field. Include the usercode if it is different from yours. Include the family name if the code file resides on a family other than those listed in your current FAMILY statement.

For example, you might enter *(SALES)OBJECT/SALES/PROJ ON APPS* to execute the code file OBJECT/SALES/PROJ residing under the SALES usercode on the APPS disk family. To display your current FAMILY statement, enter *FAMILY* in the *Action* field.

2. If the code file needs parameters, enter them in the *Parameter* field. For example, the A Series Mail System code file, OBJECT/MAIL, needs a blank parameter to run. One way to run the Mail System is to enter **OBJECT/MAIL* in the *Task name* field, and two quotation marks separated by a blank (" ") in the *Parameter* field. If a parameter contains only digits, it is treated as a numeric value unless you indicate that it is an alphanumeric value by entering X (or any nonblank character) in the field labeled *The parameter should be treated as alphanumeric*.
3. If the task needs a task value, enter it in the *Value* field.

For example, the system utility that checks for new electronic mail or news, OBJECT/CHECKMAIL, performs differently depending on the task value assigned to it. A task value of 2 causes CHECKMAIL to report whether you have new mail and new news. On the RUN screen, you can enter **OBJECT/CHECKMAIL* in the *Task name* field, leave the *Parameter* field blank, and enter 2 in the *Value* field.

- If you need to equate files or assign task attributes, enter *X* (or any nonblank character) in the *File equations are needed* field or the *Task attributes are needed* field, as appropriate.

Making File Equations

If you enter a name in the *Task name* field on the RUN screen, the system displays the FILEEQUATE screen; see Figure 4-4.

```

FILEEQUATE - FILE EQUATIONS FOR A TASK                                04:43 PM
Action: [ Home GO PArEnt ]                                           (Press SPCFY for Help)
-----
File internal name [ ]
Equation           [ ]
-----
File internal name [ ]
Equation           [ ]
-----
More file equations [ ] (any non-blank means yes)
-----
    
```

Figure 4-4. FILEEQUATE Screen

This screen provides prompts for two file equations. *File equation* is a means of specifying the values of file attributes to be used when you compile or execute a program. For example, assume you want to run the program OBJECT/PROG. This program requires an input file that is coded in the program with the name INFILE. However, the file you want to use as the input file actually exists on your system as (JACOB)INPUT/DATA on the disk family ORDSPACK. File equation enables you to associate the internal name INFILE with any actual file on the system. In this case, enter *INFILE* in the *File internal name* field, and *(JACOB)INPUT/DATA ON ORDSPACK* in the corresponding *Equation* field.

To make more than two file equations, enter any nonblank character in the *More file equations* field and press the XMIT key. The system displays another blank FILEEQUATE screen. You can make any number of file equations.

For information on file attributes, see the *A Series File Attributes Programming Reference Manual*. For information on the syntax for file equations, see the *A Series Work Flow Language (WFL) Programming Reference Manual*.

Making Task Attribute Assignments

If you enter *X* (or any nonblank character) in the *Task attributes are needed* field on the RUN screen, the system displays the TASKATTR screen. Figure 4-5 illustrates this screen.

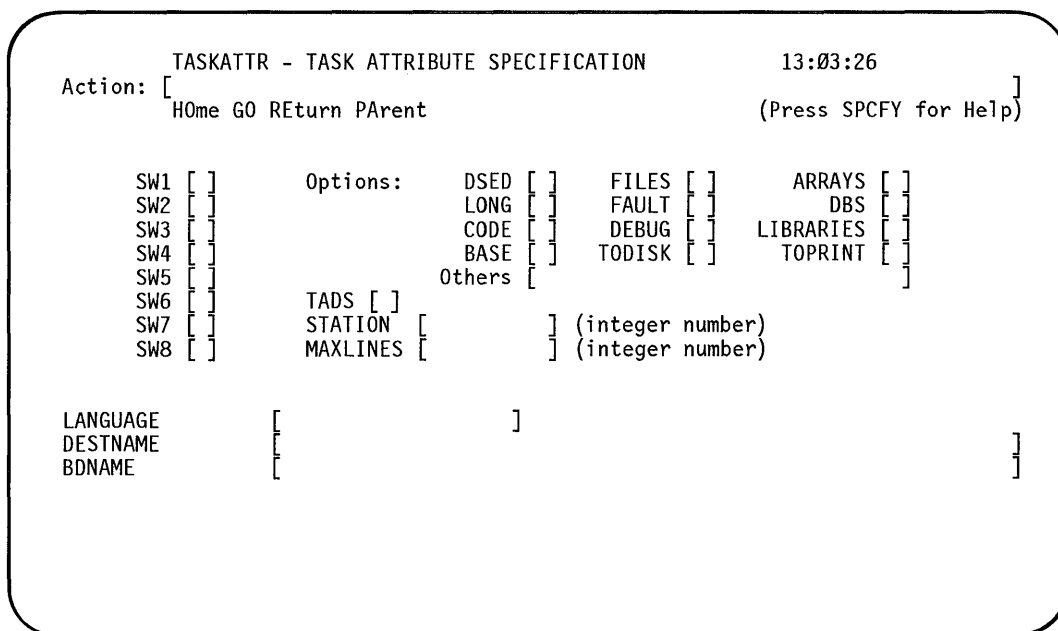


Figure 4-5. TASKATTR Screen

You can use the TASKATTR screen to assign values for a number of task attributes or task attribute options. A task attribute is an element that describes and controls various aspects of the execution of a task. The TASKATTR screen can be divided into several regions, as described in the following paragraphs.

System Switch Values

The attributes SW1 through SW8 define the values of the system switches. To enable any of these switches, enter *X* in the appropriate field. System and user programs can consult the values of these attributes during execution and make decisions based on those values.

Program Dump Options

You can use several options to specify the items of a process stack that are to be dumped and analyzed if a program dump occurs. A *process stack* is a region of memory that the program initiated by the task uses as a general purpose work area. When a task terminates, you can examine various items of the process stack to help you diagnose problems if they occur.

Several options on the TASKATTR screen are related to program dumps. Most of these options are effective only if a program dump occurs. To enable any of these options, enter *X* in the corresponding field. These options are as follows:

Option	Effect When Enabled
DSED	Causes a program dump if the program is discontinued (with a DS system command) for any reason other than a program fault.
LONG	Prevents arrays from being segmented—that is, divided up. This option affects only ALGOL, FORTRAN, or FORTRAN77 programs.
CODE	Dumps the code-segment dictionary of the task.
BASE	Dumps the contents of the stack base and of the process information block (PIB).
FILES	Dumps information about the files in the stack, including the contents of disk file headers.
FAULT	Causes a program dump if the program ends abnormally due to a program fault, such as a segmented array or a divide-by-zero error.
DEBUG	Executes special compiled-in debugging code if used by the program. This option affects only COBOL74 or FORTRAN programs.
TODISK	Sends the program dump to a disk file.
ARRAYS	Dumps the contents of all arrays used by the program.
DBS	Dumps the contents of the database stack if the database is used by the program.
LIBRARIES	Dumps all libraries associated with the stack. The output from this option includes an analysis of all library-related information, including library templates and directories.
TOPRINT	Sends the program dump to a printer.

Program dumps are described in more detail in the *A Series ALGOL Programming Reference Manual, Volume 1: Basic Implementation*.

Other Options

You can enable the following task attribute options by entering them in the field marked *Others*:

Option	Effect When Enabled
AUTORM	Causes duplicate files for this task to be removed automatically.
BACKUP	Causes printer backup files to be sent to disk regardless of the setting of the LPBDONLY or CPBDONLY operating system options.
BDBASE	Causes the task to assume some of the characteristics of a job, including producing a job summary and printing backup files when the task terminates.
NOSUMMARY	Prevents the job summary from being printed if the task produces no printer backup files. However, the job summary is printed if the task terminates abnormally.

continued

Running Programs from MARC

continued

Option	Effect When Enabled
PRIVATELIBRARIES	Dumps any private libraries used by the task. The output from this option includes an analysis of all library-related information, including library templates and directories.

Attributes Affecting Testing, LSNs, Printing, and Language

You can assign values to the following task attributes – except TADS – by entering the appropriate value for each:

Attribute	Function
TADS	Invokes a Test and Debug System (TADS) test session for a previously specified program. To enable this attribute, enter <i>X</i> (or any nonblank character) in the associated field. For more information about TADS, refer to the <i>A Series ALGOL Test and Debug System (TADS) Programming Guide</i> , the <i>A Series COBOL ANSI-74 Test and Debug System (TADS) Programming Guide</i> , or the <i>A Series FORTRAN77 Test and Debug System (TADS) Programming Guide</i> .
STATION	Overrides the logical station number (LSN) of the data comm terminal from which the task is initiated. The LSN value is assigned to any remote files used by the task.
MAXLINES	Specifies the maximum number of lines that can be printed by the task.
LANGUAGE	Specifies the natural language under which the task is run. The default is the current MARC session language. The MultiLingual System (MLS) can use this attribute to determine the language in which output messages are displayed.
DESTNAME	Assigns a station name as a destination for printer backup files. This attribute can be useful at sites where some printers are connected to data comm lines.
BDNAME	Specifies the first node of the file name for printer backup files. The default value is BD.

Running a Task from the Action Field

You can also run a task from the *Action* field of a MARC screen simply by entering the RUN command. If the RUN command is too long for the field, enter *CO*. Doing so displays a full screen in which you can enter the RUN command.

You must enter the complete RUN command; no abbreviations are accepted. Further, you must use the full code file name of the program you are running, including the usercode if it is different from yours. In MARC, for example, you might enter the command RUN OBJECT/MYPROG, for which the CANDE equivalent is R MYPROG.

Displaying the TASKSTATUS Screen

When you initiate a task from MARC with either the START command or the RUN command, the following message appears on the status line:

Task initiated.

Next, the system displays the following information:

If no input is entered, the final task message will be displayed when the task terminates.

At this point you can check the status of the task by entering TA in the Action field. Doing so causes the TASKSTATUS screen to be displayed. This screen displays the status of the task, together with any messages relating to the task as they are generated by the system. When you are on the MARC/1 dialogue, messages for any jobs that you initiate also appear on this screen.

Whenever the TASKSTATUS screen is displayed, it is automatically updated every four seconds provided a new message has been received for the task or a job. Messages are received whenever the task—or a job—becomes scheduled, becomes ready, enters a waiting state and is waiting on an external event, terminates, or displays a message.

Figure 4-6 presents an example of the TASKSTATUS screen.

```

TASKSTATUS - TASK STATUS (AND JOB MESSAGES)           05:15 PM
Action: [Return                                     ]
        Home GO Return Comnd SPlit                   (Press SPCFY for Help)
-----
Task= 8755 (INVNTRY) PARTS ON ORDSPACK
Parameter= AVERAGES
Task status= WAITING ON AN EVENT

Elapsed= 11.009           Processor= 0.015           I/O= 0.058
                           max= 1.000               max= 1.000
11:52 8755/8758 BOT (INVNTRY)PARTS ON ORDSPACK
11:52 8758 DISPLAY: COUNTUP STARTED
11:52 8758 ACCEPT: ENGLISH OR METRIC
==> 8758:AX METRIC
11:56 8758 GOING
11:56 8758 DISPLAY: OUTPUT IN METRIC TONS
11:56 8758 DISPLAY: COUNTUP FINISHED
11:56 8758 ACCEPT: AX TO PRINT FINAL RESULTS
    
```

Figure 4-6. TASKSTATUS Screen

Occasionally the message (*Press SPCFY for Help*), displayed in reverse video, overwrites some of the available actions listed on line 3. This message appears, for example, when many actions are available, but one you have entered is invalid. (The

Running Programs from MARC

TASKSTATUS screen is the only MARC screen on which this situation can arise.) To request that all available actions be redisplayed, enter a blank in the *Action* field.

Screen Display

The top portion of the TASKSTATUS display includes the following information:

Item	Meaning
Task	The mix number and the name of the task.
Parameter	The value of any parameter associated with the task.
Task status	The current state of the task, for example, SCHEDULED (temporarily prevented from starting), READY, WAITING ON AN EVENT (such as waiting for a processor to become free), or TERMINATED.
Elapsed	The total elapsed time in seconds since the task was initiated.
Processor	The total time in seconds that the task has been executing code.
max	The maximum amount of processor time that can be used by the task. This value is displayed only if a maximum limit has been set for the task.
I/O	The total time in seconds used by the task for input and output processing.
max	The maximum amount of I/O time that can be used by the task. This value is displayed only if a maximum limit has been set for the task.

The rest of the screen displays the system messages generated for this task. On the MARC/1 dialogue, the display includes job messages, messages from the ODT, and messages from other stations. Only the 11 most recent messages are displayed.

Entries in the Action Field

You can enter *MSG* in the *Action* field to display all messages received for this task or previous tasks. See Section 6, "Using Commands," for more information on the *MSG* command.

Entering an *ON* command allows you to switch to another MARC dialogue where you can run another task in parallel to the current task. If you enter an *ON* command from the TASKSTATUS screen on MARC/1 and the task is still active, you can display the TASKSTATUS screen of MARC/1 while you are in a different window. You have this ability because, by default, messages received on MARC/1 are not suspended. If you do not want to display MARC/1 tasks from a different screen, you can either access a different screen before entering the *ON* command, or suspend MARC/1. See "How MARC Stores Messages" in Section 5, "Using COMS Windows and Dialogues," for more information.

You can cause the screen to be updated with the latest information by transmitting a blank in the *Action* field. Several other options for tracking task progress are available, depending on the task state and the privilege status of your usercode. Only those actions that are currently available to you appear on the screen.

Available Actions

The following actions, also called *action words*, might appear on line 3 of the TASKATTR screen:

Action	Effect
HOME	Displays the MARC home menu.
GO	Displays the screen that has the specified screen name.
RETURN	Returns the last screen displayed before you switched to the TASKSTATUS screen.
COMND	Initiates commands. You can enter <i>GO</i> followed by a command to execute any command, provided the command does not try to initiate a task while the current task is still running. Entering <i>CO</i> causes the COMND command input screen to appear, prompting you to enter a command. See "Entering Commands" in Section 6, "Using Commands," for information on the COMND screen.
SPLIT	Ends the session for the current dialogue and starts a new session. The system discards information for the task and lists the backup files for the session. This action is available only if the task has completed.
TASK	Switches to the task window for the task. See "Displaying the Task Window" later in this section for more information. This action is available only if a task window has been opened for the task.
VIEW	Displays the TASKVIEW screen, which lists the complete task structure of the current task. This structure includes all the tasks initiated by the current task, along with their mix numbers. If the current task was itself initiated by another task, the originating task is also listed. If the originating task initiated tasks in addition to the current task, the display also includes those tasks. This action is available only if the originating task initiates other tasks and the other tasks are still active. For more information on this screen, see "Displaying the TASKVIEW Screen" later in this section.

System Command Functions Available as Actions

The following actions, presented alphabetically, correspond to system commands. See the *A Series System Commands Operations Reference Manual* for information on these commands.

Action	Effect
AX	Passes text to a task requesting an ACCEPT message.
CU	Displays the memory usage of the task.
DP	Forces a dump of the task.
DS	Terminates the task.
FA	Changes the file attributes of the task to conform to required values, providing the attributes can be changed after task initiation. Use FA only when opening files, not when closing files. FA is valid only for tasks and does not apply to an NF condition, described later in this list.

continued

continued

Action	Effect
FM	Restarts a program that has been suspended by an unsuccessful attempt to open a file for which the FORMID attribute is set.
FR	Specifies that the input tape reel just read is the final reel of an unlabeled tape file.
HI	Causes the exception event of the task to occur. The exception event is a specific event that can be predefined for the program initiated by the task. For example, the event might display how much of a file has been transferred in a file-copying operation.
IB	Displays the requested instruction block for a job. The instruction block contains instructions to the system operator about running the job.
IL	Indicates that the file requested by a program resides on the specified unit, regardless of the information on the unit label.
NF	Indicates that a file that cannot be found is to cause the task to terminate.
OF	Indicates that a file that cannot be found is to be considered an optional file.
OK	Reactivates a task that was suspended by an ST action.
OT	Displays the contents of the specified D2-stack cell for the task. For a compiler task, enter only the mix number in the <i>Action</i> field to display the sequence number of the source record being processed by the compiler, with the number of syntax errors detected.
QT	Terminates printing of a backup file.
RM	Removes the disk file specified in a DUP LIBRARY message so that a new file can be created.
ST	Temporarily suspends the task. Use the OK action to reactivate the task.
TI	Displays the current values of the time accumulators for the task in units of hours, minutes, and seconds.
UL	Assigns to the task an unlabeled file located on the indicated unit.
Y	Displays the current status of the task.

Displaying the TASKVIEW Screen

You can enter *V* or *VIEW* on the TASKSTATUS screen to display the TASKVIEW screen, provided that this action is listed on the TASKSTATUS screen. The TASKVIEW screen displays the complete task structure, enabling you to determine which tasks were initiated by the original task. The mix numbers and names of each task are displayed, along with additional information. Figure 4-7 presents an example of a TASKVIEW screen.

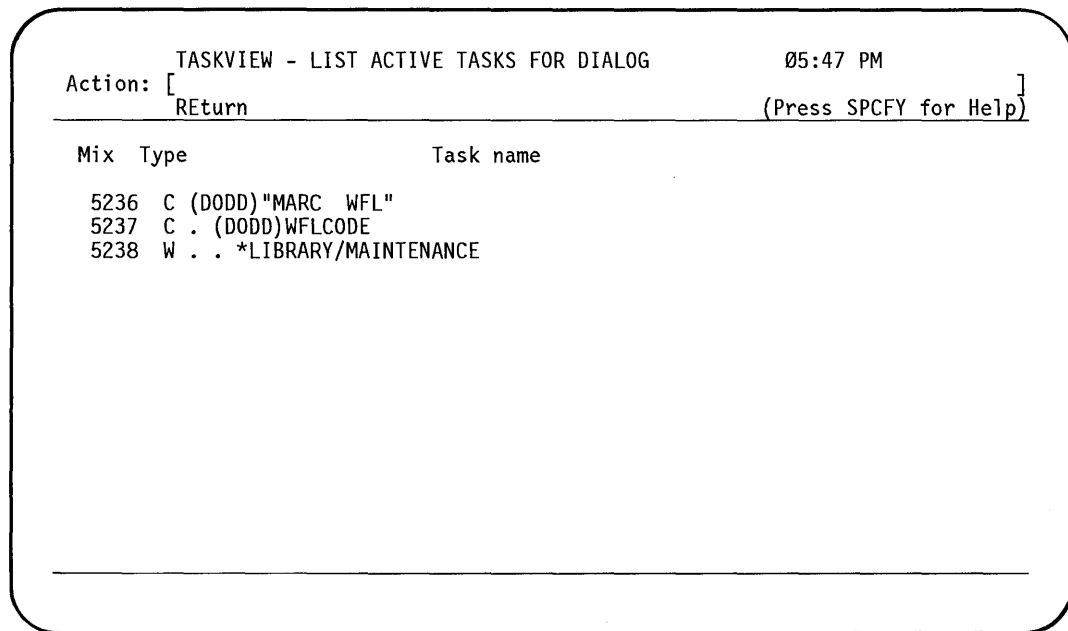


Figure 4-7. TASKVIEW Screen

You can enter any of the following items in the *Action* field of this screen:

Action	Effect
Return	Causes the TASKSTATUS screen to be redisplayed.
<mix number>	Displays the status of the task with the corresponding mix number. Use this action to see the status of any task initiated by the original task.
plus (+)	Scrolls one page forward through the list of active tasks.
minus (-)	Scrolls one page backward through the list of active tasks.

If the original task terminates while you are on the TASKVIEW screen, the system automatically redisplay the TASKSTATUS screen.

Displaying the Task Window

Some tasks open one or more special files known as *remote files*. Tasks that open remote files are typically tasks that interact with the user while they are running. For example, certain utilities available through MARC, such as the Interactive Datacomm Configurator (IDC) utility, open remote files in MARC. When a task opens a remote file, a special window called a *task window* opens for the task.

In most cases, when the task opens a remote file, MARC automatically displays the task window. The current screen disappears and MARC displays the following message:

Task Window Enter ?MARC for task status.

Running Programs from MARC

If the program initiated by the task writes to the remote file, the messages appear in the task window. If you transmit text in the task window, MARC interprets that text as input to the remote file. The only exceptions are the ?MARC command and primitive commands—that is, those commands that are prefixed with two question marks (??).

To return to the TASKSTATUS screen, enter ?MARC. To return to the task window, enter TASK in the *Action* field of any screen.

For more information on remote files, see the *A Series Communications Management System (COMS) Configuration Guide*.

Standard Tasks

When you are on the task window and the task has been suspended for some reason, the following message appears on the status line:

```
Task Window Task suspended. Enter ?MARC.
```

Enter ?MARC to return to the TASKSTATUS screen and view the displayed messages, which should indicate why the task is suspended.

Ended Tasks

When the task closes the remote file, the task window is closed. If you are on the task window when the task terminates, MARC automatically redisplay the originating screen if the value of the AUTOSWITCHTOMARC task attribute is TRUE. Otherwise, you must press the SPCFY key to redisplay the originating screen.

Remote Window

In some cases, when the task opens a remote file, MARC does not automatically display the task window. This situation occurs when you submit a WFL job through the START command and that job initiates a task that opens a remote file. When the task opens the remote file, the following message appears on the status line:

```
>>> Message received on MARC/1 <<<
```

In addition, a message similar to the following appears on the TASKSTATUS screen:

```
12:54 4280 Remote window REM0001 OPEN. INTNAME=DCOM. User=ACCTS.  
Program=*OBJECT/SALES/REPORT.
```

Note the name of the remote window (REM0001 in this case). You can switch to this window and display the messages from the program by entering an ON command that identifies the window name—for example *ON REM0001*.

The system automatically redisplay the TASKSTATUS screen when the task closes the remote window. You can also return to the TASKSTATUS screen by entering ?ON MARC.

Monitoring the Progress of a Task

As the task progresses, the system displays messages on the status line to tell you what stage the task has reached. The most common messages have to do with task initiation, suspension, and completion.

Task Initiation

Whenever a task is initiated, the following message appears on the status line:

```
Window MARC/n Task initiated.
```

When a task is initiated, the screen that initiated the task is redisplayed with **TASK** as one of the available actions. The *Action* field is prefilled with *TAsk*, and the cursor moves to that field. The following message appears at the bottom of the screen:

```
If no input is entered, the task window will appear when opened or the
final task message will be displayed when the task terminates.
```

If the task is a **COPY** or **ADD** operation, the message is the following:

```
If no input is entered, the final task message will be displayed when
the task terminates.
```

Normally, if the task opens a remote file, you are immediately switched to the task window. However, if the task does not open a remote file or the task is suspended or delayed for some reason, you can enter any available action or choice that does not attempt to initiate another task. Transmitting the **TAsk** action takes you to the **TASKSTATUS** screen, or to the task window if it is open.

If you leave the screen that initiated the task active and enter no further input, upon task termination, the final task message received for the task appears at the bottom of the screen. If no message is received for the task, the following message appears:

```
No task message received for task.
```

If you are on the **TASKSTATUS** screen when the task window is opened, you are automatically switched to the task window.

Entering Tasking Mode

Tasking mode is the name given to the mode of operation in which you monitor the progress of your task and interact with it. You enter tasking mode whenever you display the **TASKSTATUS** or **TASKVIEW** screen, or interface with your task through the task window. You leave tasking mode when you perform other MARC functions.

Various actions and commands enable you to enter and leave tasking mode and to perform functions within it. Whether you are in or out of tasking mode, MARC displays messages on the status line to keep you informed on the general progress of your task.

TASK Action

This action appears on most MARC screens once a task has been initiated by a dialogue. It appears on the TASKSTATUS screen only when a task window is currently open for the task. If you enter this action when a task window is currently open for the task, the task window reappears on your screen. If no task window is currently open for the task, the TASKSTATUS screen is displayed.

Once a task has terminated, entering *TASK* displays the final TASKSTATUS screen for the task. The contents of this screen are lost only when you initiate another task. Messages for a previously run task might still be available through an MSG command.

You can also use standard screen actions to display the TASKSTATUS screen. Entering *GO TASKSTATUS* invokes the TASKSTATUS screen for the task that was last initiated. If no task has yet been initiated on the current MARC dialogue, all fields on the TASKSTATUS screen are blank and the following message appears:

No task has been initiated on this MARC dialogue

At appropriate times, the PRev and REturn actions also display the TASKSTATUS screen.

VIEW Action

When you enter *VIEW* in the *Action* field on the TASKSTATUS screen, the system displays the TASKVIEW screen. Entering a valid mix number on this screen displays a TASKSTATUS screen for the specified task. Entering *RETURN* in the *Action* field on the TASKVIEW screen redisplay the TASKSTATUS screen for the original task.

?MARC Command

Entering *?MARC* on the task window accesses the MARC window and displays the TASKSTATUS screen for the task that opened the window.

If the task window closes but the task itself has not completed, the MARC window appears automatically. However, before any MARC screen appears, you are prompted to enter *?MARC*. This prompt enables you to view the final contents of the task window and prevents MARC from processing input that was intended for the task. After you enter *?MARC*, the TASKSTATUS screen is displayed if the task is still active. However, if the task has completed, MARC redisplay the last screen displayed outside tasking mode.

Exiting Tasking Mode

The HOME, RETURN, GO, and COMND actions on the TASKSTATUS screen enable you to exit tasking mode to perform other MARC functions. While the task is running, you can perform any MARC function other than MARC functions that require the initiation of another task. (You can perform such functions by using the ON command to switch to another MARC dialogue.) The RETURN action returns to the last screen displayed outside tasking mode.

Task Suspension

If a task becomes inactive and requires an action from you to continue, the following message appears on the status line:

Task suspended.

The TASKSTATUS screen shows the specific cause of the suspension and enables you to attend to the requirements of the task. If you are on neither the TASKSTATUS screen nor the task window, enter the TASK action on an appropriate screen to display the TASKSTATUS screen; if you are on the task window, enter ?MARC.

Task Termination

When the task has terminated, the following message appears on the status line:

Task terminated.

Note: *The message Task terminated does not necessarily mean that the task completed successfully. The TASKSTATUS screen displays the final status of the task. Enter TA in the Action field to display the TASKSTATUS screen.*

You can view the task messages in any of the following ways:

- If you are on a screen other than the TASKSTATUS screen or the task window, enter the TASK action on an appropriate screen to view the final task status.
- If you are still on the screen that initiated the task and have not entered an action, the final task message (if any) for the task is displayed at the bottom of the screen. In many cases—for example, during a simple COPY operation—the display of this message can make the display of the TASKSTATUS screen unnecessary.

The MARC window appears automatically if the following sequence of events occurs:

1. You are on the task window.
2. The task window closes.
3. MARC detects that the task itself has terminated.

If the task was initiated directly by MARC—for example, through a RUN statement, a UTIL command, a menu selection, or a form that generated such a command—and if the task set the task attribute AUTOSWITCHTOMARC to TRUE, MARC immediately returns the last screen displayed outside tasking mode.

If the task does not set AUTOSWITCHTOMARC to TRUE—the default value is FALSE—you are prompted to press the SPCFY key. Doing so enables you to view the final contents of the task window before returning to a MARC screen. After pressing the SPCFY key (or any other key), the last screen displayed outside tasking mode appears. The final task message (if any) is displayed at the bottom of this screen.

Providing a Final Task Message

The following information is important for those writing programs, utilities, or compilers that are to be run from MARC.

When a program, utility, or compiler is run as a task through MARC, the final message received by MARC for that task is at times displayed on the bottom two lines of a screen, usually the screen from which the task was initiated. This message, if meaningful, can make running the task easier in MARC.

All messages received for a task are displayed on the TASKSTATUS screen. However, ensuring that the final message indicates the general outcome of a task can sometimes enable the user of the task to bypass the display of the TASKSTATUS screen – for example, *DISPLAY: TEXT FILE 'DOC/DESIGN' TRANSLATED OK*.

For interactive programs, displaying an appropriate message once a user stops entering data – rather than writing the message to a remote file – can enable the user to be switched immediately back to the screen that initiated the task, provided the task attribute AUTOSWITCHTOMARC is set to TRUE.

MARC displays the final task message in the following format:

```
From task: <first line of final message up to 67 characters>  
<second line of final message, if required, up to 74 characters>
```

When necessary, MARC attempts to break the message into two lines at a blank character. The blank is placed on the first line. The message is truncated, if required, to fit into this format. If the message results from a program DISPLAY statement, the eight characters *DISPLAY:* are included as part of the first line.

The final task message can be either a system message – for example, *PK18 *FILE/DATA REMOVED ON PACK* – or a message generated by a program DISPLAY statement. If the final task message is to be generated by a DISPLAY statement, ensure that the DISPLAY statement is executed in the program after any statement that can cause a system message to be generated – for example, after an ALGOL CLOSE statement with the LOCK option. To suppress the display of the final task message on an ODT, set the task attribute DISPLAYONLYTOMCS to TRUE.

Section 5

Using COMS Windows and Dialogues

MARC runs under the control of the COMS message control system (MCS). A *message control system* is a system software program that controls the flow of messages among terminals, application programs, and the operating system. COMS is a general MCS that supports a network of users and provides them with an online interface to the system.

A distinctive feature of COMS is that it supports multiple windows at each terminal. A *window* is the connection between a terminal and the program that is communicating with the terminal. COMS enables you to use multiple windows at a single terminal. This feature means that you can move from one program environment to another while processing continues uninterrupted. For example, your system might have a MARC window, a CANDE window, and one or more windows providing access to application programs that run at your site. An application program need not run in a window, but if the program does run in a window, it is usually easier to access.

Additionally, COMS enables you to access separate copies of each window at your terminal. Each copy of a window is called a *dialogue*. Within each window there are up to eight dialogues, numbered 1 through 8. Therefore, you can have jobs running in several different dialogues simultaneously. For a detailed explanation of the concept of COMS windows, see the *A Series Communications Management System (COMS) Operations Guide*.

Certain windows in COMS are intended for communication with other MCSs, such as CANDE. MARC runs in the window called the *MARC window*. If your terminal is running under COMS, dialogue 1 of the MARC window is usually the active dialogue when you turn on the terminal. The screen initially displayed is the LOGON screen for MARC, as described under "Logging On" in Section 2, "Starting to Use MARC."

Entering COMS Commands

You can enter commands to COMS in the *Action* field of any MARC screen that accepts commands. Your usercode must have control-capable status, as described in Section 7, "Controlling System Security."

You can enter commands to COMS from the MARC window and from other windows. When you enter a COMS command from a window other than MARC, you must precede the command by a question mark (?). At the MARC window you can usually omit the question mark. For example, you can omit the question mark from a COMS command entered at any MARC screen that includes *COMnd* as one of the actions on line 3.

If you enter a COMS command without the preceding question mark, any output from the command is displayed on the OUTPUT command output screen. If you precede the COMS command with a question mark, the output overwrites part of the MARC

Using COMS Windows and Dialogues

screen. To restore the screen to its original appearance, use the CTRL-0-0-XMIT key sequence.

If you are at a window other than MARC, that window might provide a command that has the same form as a COMS command, including the preceding question mark. In this situation, you can precede the command with two question marks (??) to distinguish it from the COMS command.

Note: *When you enter any of the COMS commands READY, SAVE, or MOVE in the MARC Action field or on the COMND screen, you must include the word STATION as in READY STATION E638 to ready station E638. This word is optional in these commands in all other situations. See “Entering Commands” in Section 6, “Using Commands,” for information on the COMND screen.*

Using Multiple MARC Dialogues

The eight dialogues available in the MARC window enable you to conduct up to eight separate MARC sessions at the same time. You need to log on to MARC only once, in dialogue 1. When you move to a new dialogue for the first time since you logged on—for example, by entering *ON MARC/2* in the *Action* field—the system initiates a MARC session in that dialogue, and displays the MARC home menu. When you return to a dialogue where a MARC session is already running, MARC redisplay the most recent screen for that dialogue. Each dialogue stores the 15 most recent screens that were displayed on that dialogue. You can redisplay these screens in reverse order by using the PRev action in the *Action* field.

MARC permits up to 65,528 dialogues to be running MARC sessions on the system at any given time. These dialogues can be distributed among any number of terminals, although no one terminal can run more than 8 dialogues at a time.

Switching between Windows and Dialogues

You can switch between windows, and between dialogues within a window, by using the COMS command ?ON. If you enter the command in the *Action* field on a MARC screen, use the form ON, without the preceding question mark. You can use the ?ON form, with the question mark, from any of the COMS windows.

MARC interprets only upper and lower case alphanumeric characters and underscore characters (`_`) as valid window name characters. If a token containing any other character appears immediately after ?ON, MARC assumes that the token or string of tokens is text to be passed to the default window.

The ON command switches between windows and dialogues as follows:

Command	Effect
ON	Selects dialogue 1 of the current window.
ON <window name>	Selects dialogue 1 of the specified window.
ON /<dialogue number>	Selects the specified dialogue of the current window.

continued

continued

Command	Effect
ON <window name>/<dialogue number>	Selects the specified dialogue of the specified window.
ON*	Selects the default window for your station.

You can establish the default window by using the station definition menu of the COMS Utility, or by using the COMS command CREATE STATION. If no default window has been established, the default window is the MARC window, dialogue 1. For information on the COMS Utility and the CREATE STATION command, see the *A Series Communications Management System (COMS) Configuration Guide*.

To determine which window and dialogue are currently active, use the COMS command STATUS from the *Action* field. MARC accepts WRU as a synonym for the simple form of the STATUS command. However, none of the options that you can specify after the STATUS command are valid with WRU. See the *A Series Communications Management System (COMS) Operations Guide* for the syntax of the STATUS command.

How MARC Stores Messages

A message sent to your terminal without a specific dialogue number is directed to dialogue 1 of the MARC window, known as the MARC/1 dialogue. Such messages include system job-related messages and messages sent from one user to another by means of the TO command in COMS.

By default, any TO message received by the MARC/1 dialogue is displayed immediately on the status line of the currently active dialogue. Because the status line is limited to 60 characters, messages might be truncated before display. You can prevent messages sent to the MARC/1 dialogue from being displayed at the currently active dialogue by entering *?SUSPEND MARC/1*. These messages are then queued at the MARC/1 dialogue for later delivery. However, it is advisable to avoid suspending the MARC/1 dialogue because all responses to control-type commands such as *?WRU* or *?WINDOWS* return through dialogue 1, even if the control-type commands are executed at another dialogue. To reinstate the automatic display of messages received at the MARC/1 dialogue, enter *?RESUME MARC/1*.

All messages received at the MARC/1 dialogue are stored so that you can retrieve and display them. Messages are stored even if the MARC/1 dialogue has been suspended. If you are on the MARC/1 dialogue, you can view the messages by entering *MSG* in the *Action* field. If you are on any other dialogue, you can use the *?ON* command to move to the MARC/1 dialogue and then enter *MSG* in the *Action* field. If you prefer to stay on your current dialogue, you can view the messages by entering either form of the following command:

```
?PASS MARC : MSG
?PASS MARC/1 : MSG
```

How a Dialogue Inherits Attributes

Each MARC dialogue that you open during a given session inherits certain default attributes from the MARC/1 dialogue. These attributes are

- The usercode (and its associated privileges)
- The accesscode and chargecode (if any)
- The COMS application session language setting
- The MARC session language setting
- The family specification for the dialogue

For more information on usercode privileges, see Section 7, “Controlling System Security.”

The usercode, accesscode, and chargecode of the MARC/1 dialogue are always shared by all the other dialogues. You can, however, override the default family specification for an individual dialogue through the use of the FAMILY command described under “MARC Commands” in Section 6, “Using Commands.” You can also override the default language specifications – COMS application language, MARC session language, or both – as discussed in Section 9, “Using MARC Internationally.”

Section 6

Using Commands

The *Action* field enables you to execute system commands directly, without passing through a series of screens. This option is convenient if you are already familiar with the commands you want to execute. If you are unsure about what command to use, you can always use a menu.

The name of a command might be different from the selection key that generates the command from a menu. For the most part, however, the commands that you can enter directly are the same as those available through menu selections.

This section outlines the commands that are available for direct entry. Appendix C, “Commands Available in MARC,” lists references to the appropriate manuals for descriptions of commands other than MARC commands. MARC commands are described in this section.

This section also provides information on how a user with SYSTEMUSER status can use the MARC option LOGOFFSTAT to control the display of session resources used during processing.

Entering Commands

To enter commands directly in the *Action* field, you must first be using a screen that lists *COmnd* as one of the available actions on line 3. The *COmnd* action is not displayed on help screens or on the TASKVIEW screen in tasking mode. To go from a help screen or the TASKVIEW screen to entering commands directly, you must first use the *REturn* action to return to a screen that displays the *COmnd* action.

You can enter most commands directly. However, if you think the command might be confused with one of the screen actions listed on line 3, precede the command with the word *COmnd*. The minimum abbreviation for *COmnd* is *C*.

If you want to use a command that is too long to enter in the *Action* field, enter *COmnd* in the field by itself and transmit. This action causes the *COMND* screen to appear on the screen. Figure 6-1 illustrates the *COMND* screen.

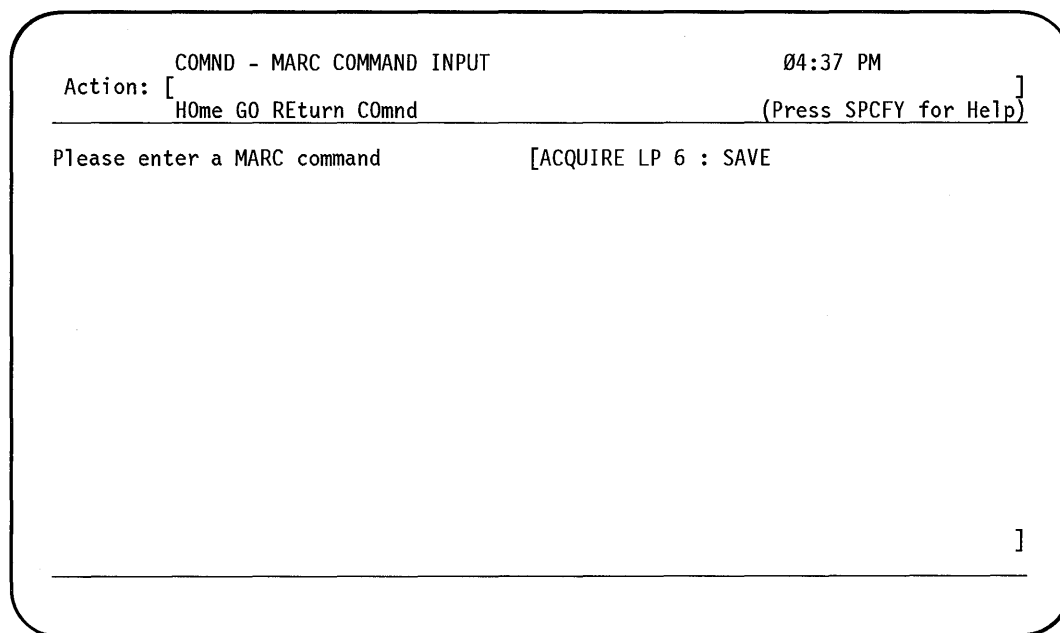


Figure 6-1. COMND Screen

A field labeled *Please enter a MARC command* occupies the middle portion of the screen. If you have already entered commands directly in the *Action* field during the current MARC session, the last command that you entered is displayed here. You can type over it if you want to execute a different command.

If you change your mind and decide to take an action other than entering a command, you can return to the screen you were using previously by using the PRev action.

If you enter a command in the command input field, the command output is displayed. If the command output is one or two lines, it is displayed at the bottom of the current screen. Otherwise, the system displays the OUTPUT screen, as described under "Displaying Output from Commands" in Section 2, "Starting to Use MARC."

If you enter a command with a syntax error, the system responds with an error message at the bottom of the screen. The incorrect command is left on the screen so you can correct it and retransmit.

Commands Available in MARC

Most of the remainder of this section discusses the majority of the commands that are available for direct entry. These commands are presented in groups: MARC commands, COMS commands, system commands, WFL statements entered as commands, Print System commands, and system utilities commands. MARC directive commands are discussed in Section 8, "Creating Custom Versions of MARC."

The other commands used in MARC are described in the following locations in this guide:

- The BYE, HELLO, and SPLIT commands, which are used to end a MARC session, are described under “Logging Off” in Section 2, “Starting to Use MARC.”
- The uses of a few of the COMS commands are outlined in Section 5, “Using COMS Windows and Dialogues.”
- Commands that are used only at an ODT running in data comm mode are described under “Configuring the ODT For Data Comm Mode” in Appendix B, “Installing MARC on Your System.”

MARC Commands

The following paragraphs describe commands that are exclusive to MARC. The commands FAMILY, FILES, LIST, NEWS, WRU, and so on are MARC commands even though they are similar to CANDE commands.

You cannot direct MARC command output to a printer.

ALLMSG (All Messages)

The ALLMSG command displays messages from the system.

Syntax

— ALLMSG —————|

Explanation

The ALLMSG command differs from the SMSG command, described later in this discussion of MARC commands, in that ALLMSG presents a rolling log containing up to 150 of the most recent messages, while SMSG presents only the 23 most recent messages. The ALLMSG command therefore enables you to examine system messages that are not recent enough to still be in the SMSG display. ALLMSG also displays messages such as beginning-of-task (BOT) and end-of-task (EOT) messages. Both ALLMSG and SMSG display the mix numbers of the tasks generating the messages, as well as the times at which the messages were first displayed.

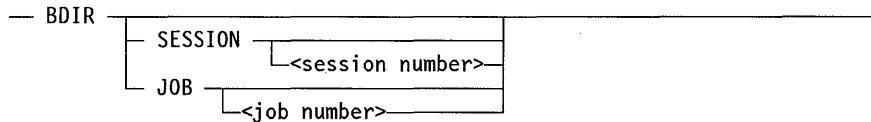
The first screen of messages displayed by the ALLMSG command contains the most recent messages. To see the older messages, scroll back in the log by using the minus (–) action.

The system messages stored for this command are lost during a halt/load, and new messages are stored when MARC is reinitiated.

BDIR (Backup Directory)

The BDIR command displays the directory of your printer backup files.

Syntax



Explanation

The directory of printer backup files includes a list of backup files created under the current session, a different session, or by a specific job.

BDIR

Displays the list of backup files for the current session in the current MARC dialogue.

SESSION

SESSION <session number>

Displays a list of backup files for the current session, or for the specified session if you include a session number. To display the current session number, enter *WRU* in the *Action* field of a MARC screen.

JOB

JOB <job number>

Displays a list of backup files for the current job, or for the specified job if you include a job number.

The session number and the job number are the same unless the job originates from WFL, in which case the job is assigned a different number.

BP (Breakpoint)

The BP command suspends execution of a program being tested under the Test and Debug System (TADS).

Syntax



Explanation

You can use the BP command only during a TADS test session. The command suspends the program being tested and returns control to TADS. For additional information, see

the *A Series ALGOL Test and Debug System (TADS) Programming Guide*, the *A Series COBOL ANSI-74 Test and Debug System (TADS) Programming Guide*, or the *A Series FORTRAN77 Test and Debug System (TADS) Programming Guide*.

BP

Suspends the task (if any) that opened a remote file, either directly or through another subtask.

BP <mix number>

Suspends the specified task. If you do not own or have access to the task, or the system fails to find the task, an error message appears.

?CONTINUE (Continue MARC Processing)

The ?CONTINUE command enables MARC processing to continue if it was suspended while waiting for a system response.

Syntax

— ? — CONTINUE _____|

Explanation

?CONT is an acceptable abbreviation for the ?CONTINUE command.

If you are waiting for a response from the MARC software or from BNA, the ?CONTINUE command displays your current MARC screen so that you can continue processing in MARC. This command is useful if you cannot wait for a response.

DIRECTIVE (Define Directives)

The DIRECTIVE command enables you to define your own commands for use with MARC. For more information on this command, see “Using Directive Commands” in Section 8, “Creating Custom Versions of MARC.”

FAMILY (Set Family Specification)

The FAMILY command displays or changes your current disk family specification.

Syntax

— FAMILY _____|
 — * _____|
 — <family specifications>

Using Commands

<family specifications>

—<target family> = —<substitute family>—————→
→ | ONLY ——————|
 | OTHERWISE —<alternate family>—|

<target family>

—<family name>—————|

<substitute family>

—<family name>—————|

<alternate family>

—<family name>—————|

Explanation

The FAMILY command either changes or displays the disk family specification for the current session. There is additionally a system default family specification (typically DISK = DISK ONLY) as well as a usercode default family specification. The usercode default specification is created by means of the MAKEUSER utility, and can be different for each usercode.

The FAMILY command is the same in MARC as in CANDE, except that in MARC you cannot abbreviate the command name to FA.

FAMILY

Displays the family name or names currently in effect for the session.

FAMILY .

Deletes any family specification made in the current session and returns to the system default specification.

FAMILY *

Deletes any family specification made in the current session and returns to the usercode default specification.

<target family> = <substitute family> ONLY

Searches the substitute family if a request is made for an existing file on the target family.

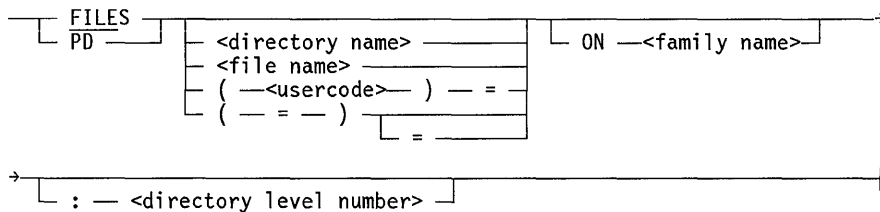
<target family> = <substitute family> OTHERWISE <alternate family>

Searches the substitute family if a request is made for an existing file on the target family. If the system fails to find the file on the substitute family, the search continues on the alternate family.

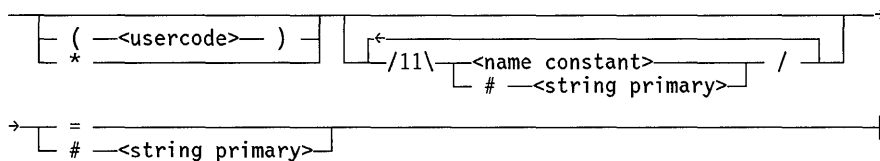
FILES (Display File Titles)

The FILES command displays the names of the files in a specified directory, or in your default directory.

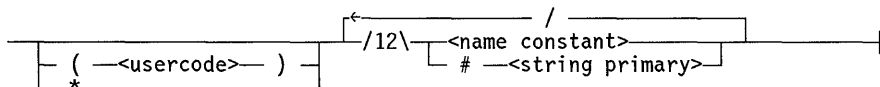
Syntax



<directory name>



<file name>



Explanation

The FILES command displays the names of files on a disk family, or displays information about a particular file. The command names FILES, FILE, and PD are all synonyms in MARC.

If no files matching your specification are found on the designated (or default) family, the system displays the message:

No file(s) were found on <family name>.

If you specify a family that the system is unable to find on the system, the response to the FILES command is

The family <family name> is missing or not in the directory.

Using Commands

FILES

Displays the names of all your files on the first (or only) disk family associated with the current MARC dialogue.

FILES <directory name>

Lists the names of files under the designated directory on the first (or only) disk family associated with the current MARC dialogue.

FILES <file name>

Displays information about the designated file if it exists on the first (or only) disk family associated with the current MARC dialogue. The display includes the following information about the file:

- File name
- File type
- Date and time of file creation
- Date and time of last access
- Date and time of last alteration
- File size in sectors
- File security type
- Archive records for the file, if any

FILES (<usercode>)=

Displays the names of all files under the specified usercode on the first (or only) disk family associated with the current MARC dialogue.

FILES (=)

FILES (=)=

Displays the names of all files under all usercodes on the first (or only) disk family associated with the current MARC dialogue. You can use the (=)= form of this command only if your usercode has SYSTEMUSER status. See Section 7, “Controlling System Security,” for more information on usercode privileges.

ON <family name>

Causes files to be listed or searched for only on the specified family.

:<directory level number>

Specifies how many levels of a directory below the directory name are to be displayed. If you specify a directory name containing x name constants and a directory level number of y , each file name in the directory is displayed out to the first $(x + y)$ levels.

Examples

The following examples illustrate selected uses of the FILES command.

- If your current FAMILY statement is FAMILY DISK = ACCTS OTHERWISE UTILS, the following command lists all the file names under your usercode on the family ACCTS:

```
FILES
```

- The following command lists information about the file named HOUSTON under the usercode SMITH on the family SALES:

```
FILE (SMITH)HOUSTON ON SALES
```

GENERATE PASSWORD

The GENERATE PASSWORD command – available on password-generating systems only – automatically generates a suggested new password.

Syntax

```
— GENERATE — PASSWORD —————|
```

Explanation

The GENERATE PASSWORD command can be used only on a password-generating system. A password-generating system has a program that generates passwords for you. If you are on a password-generating system, you cannot make up your own password; you must use a password generated by the system.

The GENERATE PASSWORD command causes the system to generate a password and display it on the bottom of your screen. If you are dissatisfied with the generated password, you can generate another one. You can then either access the CHGPASS screen or use the PASSWORD command to change your password to the new one generated by the system.

See “SHOW PASSWORD” later in this section for information on displaying the system-generated password.

For more information on password security, see the *A Series Security Features Operations and Programming Guide*.

HELP (Display Help Text)

The HELP command displays help text or help keywords from a designated help book or from the current default help book. See “Using the HELP Command” in Section 3, “Using the Online Help Facilities,” for more information about this command.

LANGUAGE (Change Language Specification)

The LANGUAGE command enables you to change the name of the *COMS application session language* – the language COMS associates with your station for this session. Or, you can use this command to change the *MARC session language* – the language MARC associates with your station for this session – or to change both session languages. The command can also list the names of both session languages.

Specifically, you can use this command for either of two purposes:

- To display the current COMS application session language and the current MARC session language
- To define a new language to be used as the COMS application session language, the MARC session language, or both

Syntax

```
-- LANGUAGE _____  
|  
| . _____  
| * _____  
| _____ <language name>  
| = _____
```

Explanation

This command displays or changes the MARC session language, the COMS application session language, or both.

LANGUAGE

Displays the MARC session language and the COMS application session language.

LANGUAGE .

Potentially changes the MARC session language, the COMS application session language, or both to the default language for the system.

The command changes the MARC session language if the default language for the system has been bound to MARC by the Message Translation Utility (MSGTRANS). If that language has not been bound to MARC, the MARC session language remains unchanged.

The command changes the COMS application session language if the command is entered from the MARC/1 dialogue. If the command is entered from any dialogue other than MARC/1, the COMS application language remains unchanged.

LANGUAGE *

Potentially changes the MARC session language, the COMS application session language, or both to the default language for your usercode.

The command changes the MARC session language if the default language for your usercode has been bound to MARC by the Message Translation Utility (MSGTRANS). If that language has not been bound to MARC, the MARC session language remains unchanged.

The command changes the COMS application session language if the command is entered from the MARC/1 dialogue. If the command is entered from any dialogue other than MARC/1, the COMS application language remains unchanged.

LANGUAGE <language name>
LANGUAGE = <language name>

Potentially changes the MARC session language, the COMS application session language, or both to the named language.

The command changes the MARC session language if the named language has been bound to MARC by the Message Translation Utility (MSGTRANS). If that language has not been bound to MARC, the MARC session language remains unchanged.

***Note:** Changes in the system default language or the default language for your usercode have no effect on sessions currently in progress.*

For information on using MSGTRANS to bind a language to MARC, see Section 9, "Using MARC Internationally."

Table 6-1 summarizes the languages referenced by those formats of the LANGUAGE command that are intended to change the MARC session language, the COMS application session language, or both. See the preceding discussion for details on each format.

Table 6-1. LANGUAGE Command Formats

Format	Language Referenced
LANGUAGE .	Default language for the system
LANGUAGE *	Default language for your usercode
LANGUAGE <language name>	Named language
LANGUAGE = <language name>	Named language

Table 6-2 summarizes the ways you can enter a LANGUAGE command to accomplish a particular goal.

Table 6–2. LANGUAGE Command Goals

Goal	Command Entered from MARC/1 Dialogue?	Referenced Language Bound to MARC?
Change the MARC session language but not the COMS application session language	N	Y
Change the COMS application session language but not the MARC session language	Y	N
Change the COMS application session language <i>and</i> the MARC session language	Y	Y

Effects of the LANGUAGE Command on Messages and the Menugraph

When you use the LANGUAGE command to change the MARC session language, MARC first determines whether messages for the named language have been bound to the MARC software. MARC examines one specific message – message 0 of the file *SYSTEM/MARC/COMMANDER, the file that is the main body of MARC – to determine if that message has been translated into the indicated language. This message reads “ok” in the original English version supplied by Unisys.

If MSGTRANS has been used to translate this message, the indicated language becomes the MARC session language. (MARC can determine whether the message has been translated even if the translation is identical to the original.) All MARC messages for this dialogue are then selected using the new MARC session language. If this message has not been translated, MARC assumes that no messages have been translated. MARC retains the language currently in use as the MARC session language.

If the search for translated messages is successful, MARC then looks for a menugraph file in the indicated language.

This file must have a title of the following form:

```
SYSTEM/MARC/MENUGRAPH/<language name>
```

This file must be located on the same family as the MARC code file.

If translated messages are found but the menugraph file is not, MARC adopts the system default menugraph as the menugraph to be used for the indicated language. Once the default menugraph has been adopted for a language, it is used whenever that language is requested until COMS is reestablished, even if the appropriate menugraph file is later added on the family. In this situation, MARC messages appear in the requested language, but the screens presented are in the default language for the system.

If the appropriate menugraph file and translated MARC messages are both available when the LANGUAGE command is entered, that menugraph file becomes the MARC session language menugraph, and the translated versions in the named language are used for all MARC messages. Online documentation is drawn from the help books associated with the menugraph file.

Effects of the LANGUAGE Command on Screens

When you use the LANGUAGE command to change the language during a MARC session, all the previous screens stored by MARC are purged and you cannot use the PRev action to redisplay screens originally displayed in the former language. However, the system stores the previous screens in the following situations:

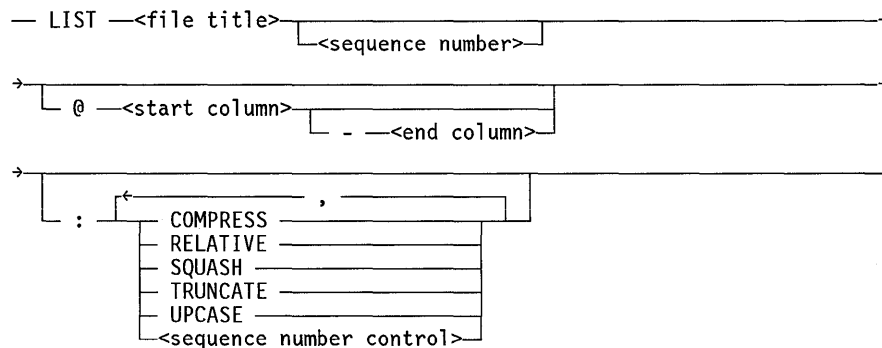
- The LANGUAGE change is unsuccessful.
- The new language specified is the same as the language that is currently active.

Note: For additional information on the LANGUAGE command, see Section 9, "Using MARC Internationally."

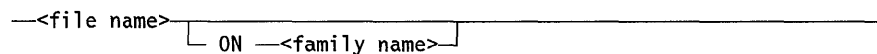
LIST (List a File)

The LIST command lists the contents of a file.

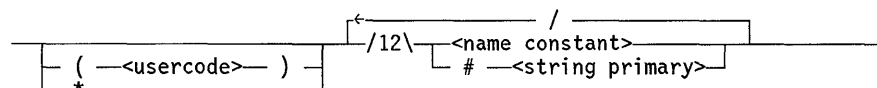
Syntax



<file title>

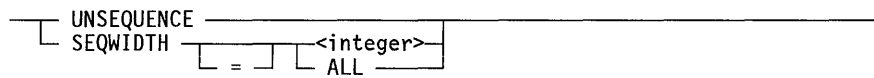


<file name>



Using Commands

<sequence number control>



Explanation

This command lists all or part of a file. Program source files, printer backup files, and files of type SEQ and TEXT are the most suitable for listing with this command. Other file types might contain control characters that affect the operation of your terminal.

The file title can begin with a usercode within parentheses or with an asterisk (*) and must match an existing file title. The file title can also include family names.

LIST <file title>

LIST <file title> <sequence number>

Displays the contents of the specified file. The optional <sequence number> construct specifies the record number at which the display is to begin.

@ <start column>

@ <start column> - <end column>

If you want to display only some of the columns, the start-column option enables you to specify the first column to be displayed, and the end-column option enables you to specify the last column to be displayed. If you omit the end-column value, the last column of the record is assumed, unless you use the TRUNCATE option, as explained later in this discussion. Column numbers can range from 1 to 180.

Example

To display columns 50 through 72 of the file FILEA, enter the following LIST command:

```
LIST FILEA @ 50-72
```

Additional Options of the LIST Command

LIST provides the following options that manipulate the text displayed on the screen:

- COMPRESS
- RELATIVE
- SQUASH
- TRUNCATE
- UPCASE
- UNSEQUENCE or SEQWIDTH

A colon (:) must precede a list of one or more of these options.

These options operate as follows:

- **COMPRESS**

Displays only one blank line between lines of text even if the source file has more than one blank line between text lines. This option deletes the blank lines in the display only; blank lines in your file are unaffected. If you enter a LIST command without this option, the file listing on the screen includes all the blank lines in the file.

- **RELATIVE**

Causes MARC to use a relative numbering system to access your file. In a file with relative numbers, each line is numbered sequentially by increments of 1, with the first line in your file displayed as relative line number 1. Relative line number 500 is thus the 500th line of the file regardless of its sequence number.

If you enter a LIST command without the RELATIVE option, MARC accesses the file by sequence number. Use UNSEQUENCE to display your file without sequence numbers.

- **SQUASH**

Reduces any group of multiple blanks on a line to a single blank. For example, if your file has more than one blank between words, the file is displayed with only one blank between each word. This option deletes the blanks in the display only; blanks in your file are unaffected. If you enter a LIST command without this option, the display includes all the blank characters in the file.

- **TRUNCATE**

Displays only one line for each record. If you enter a LIST command without this option, records longer than the display width normally wrap around onto the next line of the display. The TRUNCATE option eliminates from the display the portion of the record that would otherwise wrap around.

To display a file that is more than 80 columns wide, set this option to avoid line wraparound. Use the COL action on the subsequent display to increase your start column setting to list the truncated part of your file. In this way, you can scroll horizontally to the right to view your file. To scroll left, use COL to set the start column setting back to 1. To remove sequence numbers from the first few columns, set the UNSEQUENCE option.

- **UPCASE**

Displays the file in all uppercase letters. If you enter a LIST command without the UPCASE option, MARC displays the file without altering the case of the letters. This option changes the casing of lowercase letters in the display only; the casing in your file is unaffected.

Using Commands

- UNSEQUENCE or SEQWIDTH

These options concern the appearance of sequence numbers in the listing. A LIST command can include either UNSEQUENCE or SEQWIDTH, but not both. The options operate as follows:

- UNSEQUENCE displays text without sequence numbers, to provide more space on the screen for display. If you enter a LIST command that does not include this option, text appears to the right of a sequence number that is generated for the LIST command and might not be part of the file.
- SEQWIDTH controls the width of field in which sequence numbers appear. With this option, you specify either an integer between 1 and 8 or ALL. ALL is the default value. Specifying an integer defines the sequence number field width—that is, the number of characters to be allocated at the beginning of each line for the sequence number. When the value of the option is ALL, the system determines the sequence number field width according to the value of the FILEKIND file attribute of the file being displayed. That is, the sequence number field width equals the number of character positions required to accommodate the largest possible number for that FILEKIND.

For information about the FILEKIND file attribute, see the *A Series File Attributes Programming Reference Manual*.

The default value for each LIST command option is FALSE, with one exception: the default value of the SEQWIDTH option is ALL. To invoke a LIST command option, you need only name that option in the command, again with one exception: the SEQWIDTH option includes an optional equal sign (=) and an integer between 1 and 8 to define the number of character positions to be used for displaying sequence numbers.

Example

The following command requests a listing of FILEB with the COMPRESS option set to TRUE—that is, with only a single blank line between lines of text—and with sequence numbers in a 5-character field:

```
LIST FILEB :COMPRESS,SEQWIDTH=5
```

Displaying Printer Backup Files

To display printer backup files, you can use either the LIST command or the Backup Processor utility.

You can run the Backup Processor utility from the PS menu in MARC. The Backup Processor Utility is described in the *A Series Printing Utilities Operations Guide*.

If you use the LIST command, you can determine the file names of your backup files by using the BDIR command, described earlier in this section. When viewing printer backup files through LIST, you can use the TRUNCATE and UNSEQUENCE options, and the COL action.

MENU (Update Menugraph Version)

The MENU command displays the titles of the active menugraphs for the current session. See “Displaying the List of Active Menugraphs” in Section 8, “Creating Custom Versions of MARC,” for details on the use of this command.

MSC (Message Control)

The MSC command enables you to activate the A Series message suppression feature. With this feature, you can control the display of system messages at the system ODT and at remote terminals. This command is especially useful if your site runs jobs that produce repetitive ODT or terminal messages. You can, for example, simplify ODT and remote terminal display by suppressing announcements of each successful completion of a library maintenance file copy operation.

You can use the MSC command only if your usercode has SYSTEMUSER status. See Section 7, “Controlling System Security,” for more information on security privileges.

By default—that is, when the message suppression feature is inactive—the system displays system messages as follows:

- Messages resulting from jobs initiated at the ODT appear on the ODT screen.
- Messages resulting from jobs initiated at a *remote terminal*—that is, a terminal connected to the system by data comm lines—appear on the screen of that terminal and on the ODT screen.

With the MSC command, you can dynamically suppress or display system messages by identifying

- A type of message, such as warning messages
- A unique message characteristic, such as a specific mix number, task name, or usercode that identifies the job or jobs producing the messages to be suppressed or displayed

Note: *Unisys recommends that you specify a characteristic other than a chargecode as the characteristic to be identified. While other message characteristics are single names of up to 17 characters, chargecodes can be multilevel names.*

In addition to controlling *which* messages are displayed or suppressed, an MSC command can control *where* the suppression or display action occurs:

- At the ODT
- At remote terminals
- At both the ODT and remote terminals

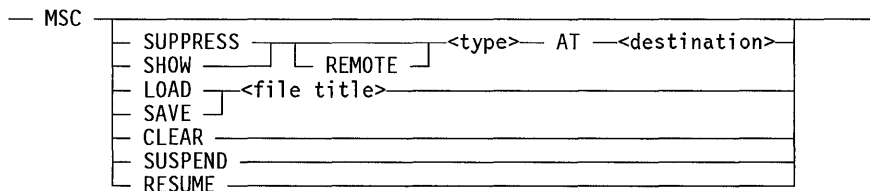
Using Commands

Specifically, the MSC command enables you to perform the following message control operations:

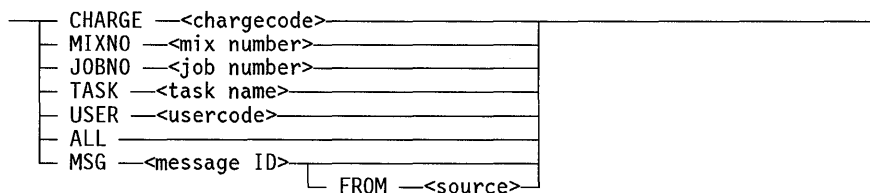
- Display the currently active MSC commands.
- Suppress the display of messages either on the ODT or on the remote terminal, or on both.
- Reactivate the display of some or all previously suppressed messages.
- Dynamically suppress or reactivate the display of messages based on specified message characteristics such as the usercode under which the job or jobs producing the messages are running, or the message ID of a specific message type.
- Temporarily suspend all active MSC commands and later reactivate those commands to resume message suppression.
- Clear all active MSC commands.
- Store one or more MSC commands as an MSC command file.
- Save the current active list of MSC commands as an MSC command file.
- Define an MSC command file as the file to be loaded automatically after a halt/load.
- Load MSC command files manually as needed.

The *A Series System Operations Guide* also provides information on the message suppression feature.

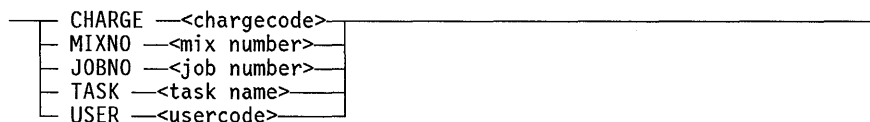
Syntax



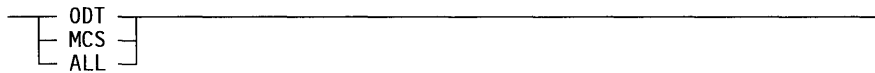
<type>



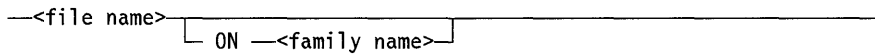
<source>



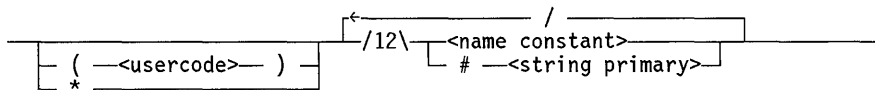
<destination>



<file title>



<file name>



Many MSC command variables—that is, items enclosed in angle brackets (< >) in the command syntax—can include “wild-card” characters. Wild-card characters are symbolic characters taken to indicate that any character or sequence of characters can appear in the corresponding position or positions in the command.

Wild-card characters are valid in the following MSC command variables:

- Variables used to identify a message type: chargecode, task name, usercode, message ID
- Variables used to identify the source of a message ID: chargecode, task name, usercode

You can use the following wild-card characters in MSC commands:

Character	Meaning
= (equal sign)	Match any sequence of characters.
~ (tilde)	Match any sequence of characters excluding the slash (/) file node separator.
? (question mark)	Match any individual character.

The wild-card mechanism enables to you perform pattern matching—that is, to instruct the system to search for messages that contain one or more common elements of a given message characteristic. For example, the following command suppresses all display of all messages from any job that has a chargecode value in which the fourth digit is 9:

```
MSC SUPPRESS CHARGE ???9 AT ALL
```

Similarly, since all warning message IDs consist of the characters *MSRWARNG* followed by a number, the following command suppresses from display at the ODT all warning messages resulting from jobs initiated from a remote terminal:

```
MSC SUPPRESS REMOTE MSRWARNG= AT ODT
```

Explanation

The MSC command enables you to control the display or suppression of system messages.

For details on type and destination specifications, see “Suppressing and Showing Messages” later in this discussion. For details on files of MSC commands, see “Creating and Using MSC Command Files” later in this discussion.

***Note:** The MSC command permits the suppression of messages that warn of potentially disastrous situations such as errors in copying during a backup operation, or disk errors during a file copy operation. Be aware of this possibility when you and your operators decide which messages are to be suppressed and where message suppression or display is to take place.*

MSC

Displays a list of the currently active MSC commands, in the order in which they were entered. The order of entry is significant, as a later command might partially or completely override an earlier command.

MSC SUPPRESS <type> AT <destination>

Suppresses the display of selected messages from jobs initiated at either the ODT or a remote terminal. Messages are selected for suppression if they are of the specified type and intended for the specified destination.

MSC SUPPRESS REMOTE <type> AT <destination>

Suppresses the display of selected messages from jobs initiated at remote terminals. Messages are selected for suppression if they are of the specified type and intended for the specified destination.

MSC SHOW <type> AT <destination>

Reactivates the display – that is, cancels the suppression – of selected messages from jobs initiated at either the ODT or a remote terminal. Messages are selected for resumed display if they are of the specified type and are intended for the specified destination.

MSC SHOW REMOTE <type> AT <destination>

Reactivates the display – that is, cancels the suppression – of selected messages from jobs initiated at remote terminals. Messages are selected for suppression if they are of the specified type and intended for the specified destination.

MSC LOAD <file title>

Appends a file of MSC commands to the currently active MSC commands on the system.

MSC SAVE <file title>

Saves the current list of MSC commands in the designated file. The current list consists of all MSC commands that are currently active on the system. To display the current list, enter *MSC* with no parameters as described earlier in this discussion.

MSC CLEAR

Removes every current MSC command. This command causes the system to revert to its default system message display procedures, outlined earlier in this discussion.

MSC SUSPEND

Temporarily suspends any message suppression currently in effect. The system follows its default system message display procedures until either a *MSC RESUME* command is issued, or a halt/load occurs.

MSC RESUME

Resumes message suppression after it was temporarily suspended by an *MSC SUSPEND* command.

Understanding How the System Uses MSC Commands

When the operating system is prompted to display a message, it reads the current list of message control commands in working memory. It then suppresses or displays the message according to the most recent applicable message directive it encounters.

The system always executes MSC commands in the order they are entered. A more recently entered MSC command thus takes precedence over an earlier command. In determining whether to display or suppress a given message, the system follows the most recently entered MSC command that applies to that message.

This procedure enables you to suppress a set of messages with one MSC command and then enter another MSC command to reactivate the display of a subset of those messages. This ability is useful when you want to suppress most – but not all – of a given set of messages. See “Combining Commands for Unique Message Control” later in this discussion for more information.

Messages that are suppressed are still available in the system. They still appear in printed job summaries, and you can list them at the the ODT by entering the system command

```
MSG ALL
```

In MARC, you can enter

```
SMSG ALL
```

The *SMSG* command is described later in this section.

Using Commands

When no message suppression commands are currently active – that is, if no message control commands reside in memory, or after an MSC CLEAR or MSC SUSPEND command has been entered – the operating system displays all job messages at their default destinations as outlined earlier in this discussion.

Entering MSC Commands

You can enter MSC commands in any of the following ways:

- Enter MSC commands directly in the *Action* field of a MARC screen.
- Access the SYS system control menu from the MARC home menu, choose the MSC selection to display the *MSC - Message Control* menu, and select the options that correspond to the operations you want to perform.
- Load one or more MSC command files. See “Creating and Using MSC Command Files” later in this section for more information on this method.

When an MSC command is entered, that command is added to the end of the list of current message suppression commands in system memory. These commands remain active until one of the following events occurs:

- A halt/load occurs.
When the operating system is halt/loaded, the part of system memory storing the MSC commands is erased.
- An MSC CLEAR command is entered.
The MSC CLEAR command erases all current message suppression commands.
- An MSC SUSPEND command is entered.
The MSC SUSPEND command suspends all MSC suppression. The suspension continues until either an MSC RESUME command is entered or a halt/load occurs.

Displaying Current MSC Commands

To display the MSC commands that are currently active on your system, enter in the *Action* field of a MARC screen

MSC

If MSC commands are active on your system, a list of all MSC suppression and display commands appears on your screen in the order they are processed. If no MSC commands are currently in use, a message to that effect is displayed.

Suppressing and Showing Messages

The SUPPRESS and SHOW options of the MSC command enable you to control message display as follows:

- The **SUPPRESS** option suppresses selected messages at designated locations.
- The **SHOW** option cancels the suppression of designated messages at designated locations – that is, it reactivates the display of selected messages. The **SHOW** option only reactivates the display of messages that were suppressed; if no messages were suppressed, the option has no effect.

An MSC command with either of these options must include a message type specification to identify the messages to be controlled and a destination specification to indicate the location or locations to be affected by the command. The command can also include the **REMOTE** qualifier.

The following paragraphs discuss your options in **MSC SUPPRESS** and **MSC SHOW** commands.

REMOTE Qualifier

You can precede the type specification with the optional **REMOTE** qualifier to limit the effect of the command to only those messages produced by a job or jobs that were initiated at a remote terminal. An **MSC SUPPRESS** or **MSC SHOW** command without this qualifier affects messages regardless of job origination.

For example, the following command suppresses the display at the ODT of all messages from programs that are executed under the usercode **OPS**, regardless of where those jobs originate:

```
MSC SUPPRESS USER OPS AT ODT
```

Messages from jobs executed under the usercode **OPS** still appear on the remote terminals from which the jobs were initiated.

In contrast, the following command suppresses the display at the ODT of messages from programs running under the usercode **OPS**, but it suppresses only a subset of those messages – specifically, those that result from jobs initiated from a remote terminal:

```
MSC SUPPRESS REMOTE USER OPS AT ODT
```

The messages still appear on the terminal from which the job was initiated.

Message Type Specifications

Table 6–3 outlines the message type specifications you can specify in an **MSC SUPPRESS** or **MSC SHOW** command.

Table 6–3. MSC Command Message Type Specifications

Type	Messages Referenced
CHARGE	All messages from programs with the designated chargecode
MIXNO	All messages from programs with the designated mix number
JOBNO	All messages from programs with the designated job number
TASK	All messages from programs with the designated task name
USER	All messages from programs with the designated usercode
ALL	All types of messages
MSG	All messages having the designated message ID

Note: An MSC command that specifies a task name takes effect only when that name matches the task name in a current mix entry.

Table 6–4 presents selected message IDs.

Table 6–4. Selected Message IDs

Message ID	Corresponding Message
MSRDIR3	<file name> COPIED FROM <source> TO <destination>
MSRDIR4	<file name> REMOVED ON <family name>
MSRDIR7	<file name> CHANGED TO <file name>
MSRDIR10	<file name> NOT ON <family name>
MSRDIR54	<number> FILES REMOVED IN <directory name>
MSRDIR85	<file name> REPLACED ON <family name>
MSRDISP3	DISPLAY: <text>
MSRDPC97	<device> PURGED
MSRFIN9	NO FILE <file name>
MSRHI13	SEG ARRAY ERROR @ <line number>
MSRIPC1	MISSING CODE FILE @ <line number>
MSRPRQ19	<device> <file name> PRINTED AND REMOVED
MSRWARNG<number>	WARNING <number> <text of warning>

If you want to reference a message that is not listed in Table 6–4, run the utility program SYSTEM/LOGANALYZER. This utility enables you to display the message ID and text of some or all of the system messages recorded in the system log file.

To initiate LOGANALYZER, select LOG on the UTIL menu. In the *LOGANALYZER Options List* field of the subsequent display, enter one of the following commands to display a list of message IDs and text:

Entry	Destination of Message Display
CON MSG	Operator display terminal (ODT)
PRINTER	System printer
REMOTE MSG	Your terminal

See the *A Series System Software Support Reference Manual* for detailed information on the LOGANALYZER utility.

For additional information on system messages, see the *A Series System Messages Support Reference Manual*. This manual discusses selected system messages generated by the master control program (MCP). For each message presented, the manual describes the cause and suggests possible responses.

When you enter an MSC SUPPRESS MSG or MSC SHOW MSG command, you can include the optional FROM qualifier to limit the selection of messages with the specified message ID to only those occurrences of that message coming from a specified source. For example, you can suppress or show only occurrences of a specified message that come from a particular job by specifying the appropriate job number in a FROM TASK qualifier following the message ID.

Table 6-5 lists the values you can specify in the FROM qualifier.

Table 6-5. Message ID FROM Qualifier Values

Value	Characteristic Common to Selected Messages
CHARGE <chargecode>	Specified chargecode value
MIXNO <mix number>	Specified mix number
JOBNO <job number>	Specified job number
TASK <task name>	Specified task name
USER <usercode>	Specified usercode

Note: *An MSC command that specifies a task name takes effect only when that name matches the task name in a current mix entry.*

For example, the following command suppresses from display on the ODT all occurrences of message MSRDPC97 resulting from all jobs, regardless of origin, with a chargecode value of 8179:

```
MSC SUPPRESS MSG MSRDPC97 FROM CHARGE 8179 ON ODT
```


Message Destination Specifications

Table 6–6 outlines the available destination specifications.

Table 6–6. MSC Command Message Destination Specifications

Destination	Meaning
ODT	Operator display terminal (ODT)
MCS	Remote terminal from which job originated
ALL	Both the ODT and the remote terminal

For example, the following command suppresses from display at the ODT messages from all jobs executed under the usercode HDS, regardless of where they originated:

```
MSC SUPPRESS USER HDS AT ODT
```

Messages from a job running under usercode HDS still appear on the terminal or terminals from which the job or jobs originated.

In contrast, the following command suppresses messages from all jobs executed under the usercode HDS, but the suppression occurs at terminal or terminals from which the job or jobs originated:

```
MSC SUPPRESS USER HDS AT MCS
```

You can include the `REMOTE` qualifier in this command if you choose, but it is unnecessary to do so: the only messages that ever appear on remote terminals are from jobs initiated from remote terminals. In either case, messages from jobs run under usercode HDS still appear on the ODT.

To suppress the display at both locations, enter

```
MSC SUPPRESS USER HDS AT ALL
```

Combining Commands for Unique Message Control

By entering `MSC SUPPRESS` and `MSC SHOW` commands in the proper order, you can suppress a class of messages but show a subset of that class.

As a small example, the following commands suppress from display at the ODT all messages from all jobs run under the usercode OPS, regardless of where the jobs originated, but displays all messages at the ODT from job number 7683 even if that job is executed under usercode OPS:

```
MSC SUPPRESS USER OPS AT ODT  
MSC SHOW JOBNO 7683 AT ODT
```

As another example, the following command suppresses the display at the ODT of all messages from jobs that originate from a remote terminal and have a message ID of MSRDIR54, but allows all such messages to appear on the remote terminal from which the job originated:

```
MSC SUPPRESS REMOTE MSG MSRDIR54 AT ODT
```

If the following command is entered subsequently, all messages from jobs initiated from a remote terminal and running under usercode OPS—including messages with a message ID of MSRDIR54—appear at both the ODT and the remote terminal, their normal message destinations:

```
MSC SHOW REMOTE USER OPS AT ODT
```

Messages with a message ID value of MSRDIR54 from all jobs initiated from remote terminals under other usercodes continue to be suppressed at the ODT.

The following example illustrates the importance of the order in which you enter MSC commands. As shown, these commands combine to suppress messages from display at the ODT messages from all jobs initiated at remote terminals under the usercode XYZ, except messages from jobs that have the task name MY/PROGRAM:

```
MSC SUPPRESS REMOTE USER XYZ AT ODT
MSC SHOW TASK MY/PROGRAM AT ODT
```

If these commands were reversed, the MSC SHOW command would have effect only if it applied to an earlier message suppression operation, and all messages from all programs initiated at a remote terminal under the usercode XYZ would be suppressed at the ODT.

See “MSC Command Examples” for additional illustrations of how MSC SUPPRESS and MSC SHOW commands can be used in combinations to achieve very specific message control.

Creating and Using MSC Command Files

The message suppression feature enables you to perform the following operations:

- Create one or more MSC command files—that is, one or more files containing one or more MSC commands that direct the operating system to suppress or display selected system messages.
- Create a default MSC command file that is to be loaded after a halt/load occurs
- Dynamically load a new file of MSC commands.
- Dynamically save the current list of MSC commands on a file.

Before using this feature, make the following decisions:

Using Commands

- Do you want to establish a default file of MSC commands to be used after a halt/load has occurred?

If you do not establish such a file, your operator must enter MSC commands manually.

- Do you need to establish several files that might be used during different times of the day?

If you establish multiple files, you can load a specific file using the LOAD option of the MSC command or the LOAD selection on the MSC screen.

- Do you just want to add or override MSC commands dynamically as your needs become apparent?

You can use this approach even if you load files that perform basic message suppression operations.

All MSC commands not specifically saved in separate files are erased when a halt/load occurs. If you decide to create an MSC command file, you have several methods of creating that file:

- Create a file of MSC commands by using an editor such as CANDE.
- Create a file programmatically.
- Enter MSC commands in the *Action* field or on MSC screens and save them by using the SAVE option.

If you choose to enter MSC commands as you need them, you can enter MSC commands in the *Action* field of any MARC screen or you can use the MI, UC, and AM selections on the MSC menu. Entering MSC commands on MARC screens and then saving them with the SAVE option adds to the already existing list of active commands. To create a completely new file, you must first use the CLEAR option.

If you choose to create the file with an editor or a program, follow these guidelines:

- The file must be either a text file or a symbol file.
- The file must contain one command per record.
- Comments, if included, must be preceded by a percent sign (%). Any input following the percent sign on the same line is ignored.
- The file should contain no MSC command that identifies a mix or job number. Any such command is ignored.
- Omit the keyword *MSC* from the MSC commands. The *MESSAGESUPPORT* function, which processes an MSC command file, does not require this keyword and rejects any command that contains it, resulting in a syntax error.

When you create an MSC command file by entering MSC commands in the *Action* field or on MSC screens and saving them by using the SAVE option, the system automatically removes the *MSC* keywords before storing the commands on the file.

To define a file as the default message suppression command file—that is, the file that the system loads automatically after a halt/load—name that file *MESSAGES/STARTUP and place it on the disk file family where the code file pointed

to by the MESSAGESUPPORT function resides. You can determine that family by entering the command

```
SL MESSAGESUPPORT
```

If this command returns just the file name, that file resides on family DISK. For information on the SL (Support Library) system command, see the *A Series System Commands Support Reference Manual*.

For example, you could use the CANDE command MAKE to create a file containing the following MSC commands you want to be automatically loaded after a halt/load:

```
SUPPRESS REMOTE MSG MSRDIR3 AT ODT
SUPPRESS REMOTE MSG MSRDIR5 AT ODT
SUPPRESS REMOTE MSG MSRDIR6 AT ODT
SUPPRESS REMOTE MSG MSRDIR7 AT ODT
SHOW USER TOM AT ODT
SHOW USER DICK AT ODT
SHOW USER MARY AT ODT
```

Note the absence of *MSC* at the beginning of these commands. The first four commands suppress the display at the ODT of four selected messages from jobs initiated at remote terminals. The three SHOW commands override the SUPPRESS commands in the cases of three particular usercodes – the usercodes of the system operators.

This sequence suppresses the repetitious display at the ODT of routine messages from jobs initiated from remote terminals, but displays on the ODT any occurrences of those messages if they result from a job submitted by an operator.

Suppose that you assign this file the name DEFAULT/MESSAGES. In addition, suppose that the code file pointed to by the MESSAGESUPPORT function resides on the disk file family named SYSTEM.

To define DEFAULT/MESSAGES as the default message suppression command file on your system, enter the following command:

```
COPY DEFAULT/MESSAGES AS *MESSAGES/STARTUP TO SYSTEM(PACK)
```

You can create more than one file of MSC message suppression commands to suit different situations. You might want to suppress one set of messages during daily operations or at system initialization, but want a different set of messages suppressed during nightly or weekend processing.

MSC Command Examples

The following examples illustrate selected uses of the MSC command.

- The following example suppresses the display at the remote terminal of messages from all programs initiated from a remote terminal and executed under the usercode OPS:

Using Commands

```
MSC SUPPRESS REMOTE USER OPS AT MCS
```

The messages still appear at the ODT.

- The next example suppresses the display at the ODT of messages from all programs, such as OBJECT/EDITOR, having a task name that begins with the characters *OBJECT/ED*:

```
MSC SUPPRESS TASK OBJECT/ED= AT ODT
```

- The following command suppresses from display at both the ODT and the remote terminal all occurrences of message MSRDIR4 resulting from jobs executed under usercode STAT:

```
MSC SUPPRESS MSG MSRDIR4 FROM USER STAT AT ALL
```

- The following command suppresses from display at the ODT all occurrences of message MSRDIR3 stemming from jobs initiated at remote terminals:

```
MSC SUPPRESS REMOTE MSRDIR3 AT ODT
```

Messages with the message ID value of MSRDIR3 are of the form

```
FILE COPIED FROM DISK1 TO DISK2
```

These messages are apt to be of more interest to the user of the program than to the operator of the system. The messages still appear on the terminal from which the job originated.

- The following command suppresses the display at the ODT of messages from all jobs that are initiated from a remote terminal and that specify chargecode XXXX:

```
MSC SUPPRESS REMOTE CHARGE XXXX AT ODT
```

These messages still appear at the terminal or terminals from which the job or jobs were initiated.

MSG (Messages)

The MSG command displays your task messages.

Syntax

— MSG —————|

Explanation

This command displays the task messages that have accumulated at the current dialogue. On the MARC/1 dialogue, job messages, messages from the ODT, and messages from other stations are also displayed.

This command differs from the MSG (Display Messages) system command, which displays system messages. To display system messages in MARC, use the ALLMSG or SMSG command, both included in this discussion of MARC commands.

The first screen of messages displayed by the MSG command contains the most recent messages. You can reach older messages by using the plus (+) action to scroll forward in the file.

To purge messages associated with a MARC dialogue, enter a BYE or HELLO command. A SPLIT command purges messages associated with MARC dialogues 2 through 8; messages associated with the MARC/1 dialogue are unaffected by a SPLIT command.

NEW (Update Menugraph or Book Version)

The NEW command makes a new version of a menugraph or help book available to users on the system.

Syntax

```

NEW — MENU —<menugraph title>
      |
      | BOOK —<book title>
  
```

Note: *Include the usercode and family name with the file title if the menugraph or help book file resides under a usercode or on a family that is different from the usercode or family of the MARC code file.*

Explanation

Use this command whenever you update an in-use menugraph or online help book. This command applies to menugraphs and help books used by MARC and other products created by means of IMG: products such as IMG itself, IDC, BNA Version 1, and any user-written menugraph (or its associated help book).

When a menugraph or help book is first accessed during a session, the system copies the menugraph or help book from disk into a cache in system memory, assigning “in use” status to the file. The cache enables the system to access the file much faster than it can from disk. The display output by the FILES command shows whether a file is marked *in use*.

You update a menugraph or help book that is currently marked *in use* by updating the file on disk. The copy of the file in system memory is unaffected until you issue the NEW MENU or NEW BOOK command. Thus, any changes made to an existing menugraph or help book are unavailable to users until a NEW MENU or NEW BOOK command is issued.

A successful NEW MENU or NEW BOOK command causes the message “ok” to appear at the bottom of the screen.

MARC responds to a NEW command for a menugraph or help book that is not currently marked *in use* by displaying the following message:

```
Not currently using book named: '<book title>'.
```

See “Making a Custom Menugraph Available to Users” in Section 8, “Creating Custom Versions of MARC,” for details on the use of the NEW command with menugraphs.

Using Commands

See the *A Series Help Utility Operations Guide* for details on the use of this command with online help books.

NEWS (Display News File)

This command enables you to display or modify the contents of the current MARC system news file. The NEWS command displays the contents of the current system news file.

Syntax

```
— NEWS _____|
      |_____|<file name>|
```

Explanation

This command displays or modifies the MARC news file.

NEWS

Displays the current news file. The most recent news is listed first; you can display older news by using the plus (+) action to scroll forward in the file.

NEWS <file name>

NEWS = <file name>

Stores the first 200 records of the specified file as the MARC news file. This is the file that is displayed when you enter *NEWS*. You can use this form of the NEWS command only if your usercode has SYSTEMUSER status.

The file that is assigned as the MARC news file must have its FILEKIND attribute equal to one of the following values:

ALGOLSYMBOL	NDLII
CDATA	NEWPSYMBOL
DASDL	PLI
DATA	SANSSYMBOL
DCALGOLSYMBOL	SEQDATA
DM	SORT
FORTRAN	TEXTDATA
JOB	

MARC attempts to read up to the first 72 characters of each record in the file, except in the case of files with a FILEKIND value of CDATA, DATA, or JOBSYMBOL. With these files, MARC attempts to read the first 78 characters. If the MAXRECSIZE attribute of the file has a value smaller than 72 characters, MARC can read only the number of characters specified by the MAXRECSIZE attribute.

For information about the FILEKIND and MAXRECSIZE file attributes, see the *A Series File Attributes Programming Reference Manual*.

PASSWORD

The following is the syntax for the PASSWORD command.

Syntax

```
— PASSWORD —<old password>—<new password>—<new password>—————|
```

Explanation

If you have been given a password that expires, it is considered an aging password. You can change an aging password on the CHGPASS screen, which is displayed automatically when your password is about to expire. If you are not on an aging-password system, you can change your password on the PASSWORD or CHGPASS screen. It is important to remember to generate a password in private to maintain the security of your files.

For more information on password security, see the *A Series Security Features Operations and Programming Guide*.

PDIR (Process Directory)

The PDIR command enables you to perform any of the following file management functions:

- Select one or more files that meet a user-defined set of criteria, including wild-card searching on file names.
- Display a defined set of file attributes for each file in a list.
- Sort a list of files according to the values of one or more file attributes.

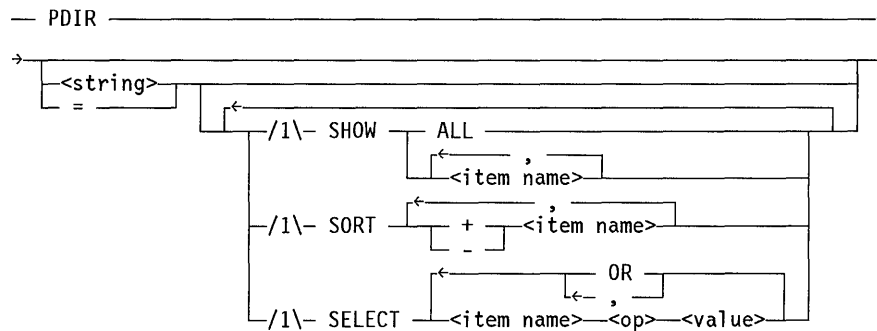
Other MARC commands such as FILES search for requested information and display each item of information that meets the search criteria as it is found while continuing to search for others. In contrast, the PDIR command completes the searching process before displaying the requested information. As a result, PDIR commands can be time-consuming.

As a safeguard against excessive consumption of system resources by PDIR commands, MARC assesses each PDIR command it receives to determine whether its processing—in addition to any other PDIR command processing currently under way—would consume excessive system resources. If so, MARC rejects the command and issues a message suggesting that you reenter it at a later time.

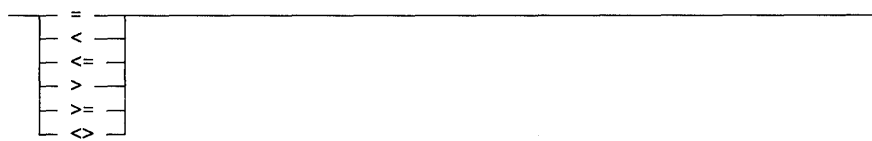
Using Commands

Note: You cannot execute the PDIR command if only one copy of MARC is running. The number of running copies is determined by COMS and always falls between a minimum and maximum number. The Program entity of the COMS Utility program enables you to display the current values of these numbers and then to modify those values. The default maximum is 2.

Syntax



<op>



Explanation

This command performs a variety of file management functions.

PDIR

Displays a default list of information about all files under the current usercode on the current disk family.

PDIR <string>

Displays information about one or more files or directories. The string can be the name of an individual file or directory, and can optionally include a usercode and a family name.

You can use wild-card characters in the string to display information about all files whose names match the string. The wild-card characters are

Character	Meaning
=	Match any sequence of characters.
^	Match any sequence of characters excluding the slash (/) file node separator.

continued

continued

Character	Meaning
?	Match any individual character.

See the end of this command description for examples of using wild-card characters.

PDIR =

Displays information about all files under the designated (or current) usercode, directory, and family name.

SHOW ALL

SHOW <item name>

Display selected information about one or more files. To display the values for all the items in Table 6–7, following, enter *SHOW ALL*. To select information to be displayed, enter one or more item names separated by commas (,).

Table 6–7 lists the item names you can use in the *SHOW* parameter. Most item names correspond to file attributes for which you can display values. Any item that is other than file a attribute – or that does not correspond exactly to a file attribute – is noted in the table. For example, the *PDIR* parameter *SHOW ALTERNATE* displays the date and time of the last alteration to the file, whereas the *ALTERNATE* file attribute value displays only the time of last alteration. For a detailed explanation of file attributes and their values, see the *A Series File Attributes Programming Reference Manual*.

If a specified item has a null value, the item is either omitted from the display or shown with a blank value.

If you omit the *SHOW* parameter, the system displays, for each file selected, the values of the *NAME*, *FILEKIND*, *RECORDS*, *SECTORS*, *CREATIONTIME*, and *LASTACCESSTIME* items.

Table 6–7. PDIR Command Item Names (SHOW, SORT, SELECT Parameters)

Item Name	Meaning
ALTERNATE†	Date and time when the file was last altered
AREALENGTH	Number of FRAMESIZE units in the file area
AREAS	Number of areas the file can allocate
AREASINUSE†	Number of areas used to store the file
AREASECTORS	AREASIZE in sectors
AREASIZE	Number of logical records in the file area

† Differs from a similar file attribute.

continued

Table 6–7. PDIR Command Item Names (SHOW, SORT, SELECT Parameters) (cont.)

Item Name	Meaning
BLOCKSIZE	Size of the I/O buffer for the file (see UNITS)
BYTES†	Number of bytes of data in the file
CREATIONTIME†	Date and time when the file was created
CRUNCHED	Indication of whether the system can reallocate the unused part of the last file area
CYCLE	Generation number of the file (for cataloging)
EXTMODE	Code format of the file, such as ASCII or EBCDIC
FAMILYNAME	Name of the disk family on which the file resides
FILEKIND	File classification, such as ALGOLCODE or SEQDATA
FILELENGTH	Size of the file in FRAMESIZE units
FILEORG†	File organization, such as relative, indexed, or KEYEDIOII
FILETYPE	Format of the file records, such as fixed-length or variable-length
FRAMESIZE	Frame size—the number of bits transferred during each I/O operation on the file
INUSE†	Indication of whether the file is currently in use by the system—that is, whether the file has been read into memory
LASTACCESSTIME†	Date and time when the file was last accessed by any user
LEVELS†	Directory level at which the file resides
MAXRECSIZE	Maximum size of the records in the file (see UNITS)
MINRECSIZE	Minimum size of the records in the file (see UNITS)
NAME†	File name (displayed by default)
NOTE	User-defined attribute, such as the text for the banner page of a printer backup file
RECORDS†	Size of the file in records
RELEASEID	Unisys software release level for the file
SAVEFACTOR	Value used by the SORT procedure and the FILECOPY utility
SECTORS†	Size of the file in sectors
SECURITYGUARD	Name of the associated guard file, if any
SECURITYTYPE	Access permission for the file, such as PUBLIC or PRIVATE
SECURITYUSE	Manner in which the file can be accessed by a nonprivileged user—IN, OUT, IO, or SECURED

† Differs from a similar file attribute.

continued

Table 6–7. PDIR Command Item Names (SHOW, SORT, SELECT Parameters) (cont.)

Item Name	Meaning
SENSITIVE	Indication that the file contains sensitive data and that its areas are to be overwritten before reallocation
TIMESTAMP	Timestamp value from the file header
UNITS	Units used for the AREALENGTH, BLOCKSIZE, MAXRECSIZE, and MINRECSIZE attribute values – bytes or words
USERINFO	User-defined attribute
VERSION	Version number of the CYCLE attribute value (for cataloging)

† Differs from a similar file attribute.

SORT + <item name>

SORT – <item name>

Displays a list of files sorted by the values of one or more item names, separated by commas. *SORT +* sorts the files in ascending order; while *SORT –* sorts the files in descending order. If you specify more than one item name, the list is sorted first by the value of the first item, then by the value of the next item, and so on.

If you omit the SORT parameter, the list is sorted first by the length of the first node of the file name, and then alphabetically by file name.

SELECT <item name> <op> <value>

SELECT <item name> <op> <value> , <item name> <op> <value>

SELECT <item name> <op> <value> OR <item name> <op> <value>

Selects files for display based on various selection criteria. You can specify selection criteria in any of the following ways:

- The command can define a single selection expression consisting of an item name, a logical operator, and a value. The system selects files that have the specified values. For example, you can select all files associated with a certain guard file.
- The command can define multiple selection expressions, separated by commas (,). The system selects only those files that have all the specified values. For example, you can select all files created after a certain date *and* greater than a certain size.
- The command can define two sets of selection criteria – each a single selection expression or a string of selection expressions – separated by the word *OR*. The system selects files that have the values defined in either one set of criteria or the other but not necessarily both. For example, you can select all files that are either created after a certain date or else greater than a certain size.

The operator in a selection expression can be any of the following:

Operator	Meaning
=	Equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
<>	Not equal to

The value can include wild-card characters if it is a character string.

Examples

The following examples illustrate selected uses of the PDIR command.

- The following example lists all files with the usercode SMITH on the default family that have names beginning with the characters *JOB*. For each file, the display includes the creation date and time, number of records, and file kind.

```
PDIR (SMITH)JOB= SHOW CREATIONTIME, RECORDS, FILEKIND
```

- The following example lists all files that reside under the current usercode on the disk named TEST and that contain the letter *A* somewhere in their names:

```
PDIR =A= ON TEST
```

- The following example lists all files under the current usercode that reside on the current disk family and that contain the letter *B* in the second position of their names:

```
PDIR ?B=
```

- The following example lists all files under the current usercode that reside on the current disk family, that start with the letter *M*, and that have at least two nodes in their names:

```
PDIR M~/=
```

- The following example lists all files under the current usercode that reside on the current disk family and that start with the letter *A*:

```
PDIR A=
```

- The following example lists all files on the current family under the current usercode with file names containing four characters. A list of file directory names containing four characters also appears:

```
PDIR ????
```

- In the following example, the system displays a list of files sorted in ascending order first by number of records and then by file name:

`PDIR = SORT + RECORDS, + NAME`

- The following example lists the name of the file on the current pack and under the current usercode most recently edited with the A Series Editor either in the current session, in an incomplete previous session, or in an externally terminated session:

`PDIR EDITOR/RECOVERY/= SORT - LASTACCESSTIME`

- The following example lists all files under the current usercode on the current pack beginning with those currently in use:

`PDIR = SORT - INUSE SHOW INUSE`

- In the following example, the system displays a list of files sorted in descending order by sector size and includes the number of sectors, the area size, and the time the file was most recently accessed:

`PDIR = SORT - SECTORS SHOW SECTORS,AREASIZE, LASTACCESSTIME`

- The following example lists all files that either have a `FILEKIND` value equal to `SEQDATA` or contain more than 5,000 records:

`PDIR = SELECT FILEKIND=SEQDATA OR RECORDS>5000`

- The final example lists only those files containing more than five records. The list is sorted by creation date, with the most recently created files shown first:

`PDIR = SORT - CREATIONTIME SELECT RECORDS > 5`

SHOW PASSWORD

The `SHOW PASSWORD` command—available only on password-generating systems—displays the next system-generated password to be used.

Syntax

`— SHOW — PASSWORD _____|`

Explanation

If you are using a password-generating system and your password is set to expire after a certain number of days, you need to know the system-generated password so you can change your password. You cannot make up your own password on a password-generating system.

The `SHOW PASSWORD` command displays the next system-generated password to use when you change your password. If you are dissatisfied with the system-generated password, you can generate another one by using the `GENERATE PASSWORD` command. See “`GENERATE PASSWORD`” in this section for more information.

You can also display the system-generated password and change your password by using the `CHGPASS` screen.

Using Commands

Once you have changed your password to the system-generated password, your preceding password is unavailable for display. You can display only the new system-generated password.

SMSG (System Messages)

The SMSG command displays recent system messages.

Syntax

```
-- SMSG [ ALL ] _____|
```

Explanation

The SMSG command provides the same function as the MSG (Display Messages) system command. That is, it displays information about the most recent messages the system has produced, presented in reverse chronological order. The display comprises approximately 23 messages. Each entry lists the mix number of the task that generated the message, the time the message was issued, and the text of the message.

If the ALL option appears in the command, the list includes any messages that were suppressed from display by the MSC command, described elsewhere in this discussion of MARC commands.

The status of your usercode affects the list produced by the SMSG command as follows:

- If your usercode has SYSTEMUSER status, the list includes the most recently produced system messages regardless of the usercode of the jobs that produced them.
- If your usercode lacks SYSTEMUSER status, the list includes only messages resulting from jobs run under your usercode.

SUSPENDUSERCODE (Suspend Usercode)

The SUSPENDUSERCODE command enables you to suspend the use of your usercode if you suspect that a security violation has occurred.

Syntax

```
-- SUSPENDUSERCODE _____|
```

Explanation

This command suspends the use of your usercode. If you feel that the security of your usercode has been violated and your usercode is being used to illegally access the system, you can use this command to stop its use immediately.

Using Commands

You can enter the ?WRU command, with the question mark included, in the *Action* field of a MARC screen, as well as on the LOGON screen. You can use the ?WRU form to verify the station you are on even before you log on.

The output of the ?WRU form writes over the three lines following the current cursor position of the current screen. To refresh the screen, press the CTRL key, press the zero key twice, and transmit.

The WRU command also displays the time, date, and station name where you last logged on. This information can help you to determine if your usercode is being illegally used to access the system. If you find that your security code has been violated, you can have your current usercode suspended, and a new one assigned. For instructions on how to suspend your usercode, see "SUSPENDUSERCODE (Suspend Usercode)" earlier in this section.

Note: *The Billing Support library adds its own messages to the end of the WRU command response, if appropriate billing options have been activated. For information on the A Series billing support facilities, see the A Series System Administration Guide.*

COMS Commands

You can enter any of the COMS commands in the *Action* field, or on the COMND screen displayed by the COMnd action. Your usercode must have control-capable status, as described in Section 7, "Controlling System Security." Appendix C, "Commands Available in MARC," contains a list of the command names and functions. The commands are documented in the *A Series Communications Management System (COMS) Operations Guide*.

System Commands

You can enter most system commands in the *Action* field, or on the COMND screen displayed by the COMnd action. Appendix C, "Commands Available in MARC," contains a list of the command names and their functions. The commands are described in the *A Series System Commands Operations Reference Manual*.

A few system commands have a slightly different syntax in MARC. If a system command requires that a numeric parameter follow the command, as in *<mix number>HI<number>*, the MARC command syntax requires that a blank appear between the command and any following parameter. For example, the HI command *123HI12* results in an error message stating that the command keyword is unknown to the system, while the command *123HI 12* is processed correctly.

The following system commands are unavailable in MARC:

Command	Meaning
AB	Automatic backup
ADM	Automatic display mode
BNAV1	BNA Version 1

continued

continued

Command	Meaning
BNV2	BNA Version 2
COMPILE STATUS	Compiler task information
FORM	Assign form identifier
LABEL	Label ODT
RO	Reset options
SP	Show print queue
TERM	Terminal

The AB, FORM, and SP commands deal with printing functions. These functions are provided by the Print System PS commands, which are described in the *A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide*.

In MARC, the only form of the PB (Print Backup) system command you can use is *PB MT <unit number>* to rewind and read a specified tape drive and send the contents to the first available printer or punch device. For information on the PB command, see the *A Series System Commands Operations Reference Manual*.

File and Message Handling Commands

The PD (Print Directory) system command is treated by MARC as synonymous with the MARC version of the FILES command, which has extended capabilities.

The MSG system command is available in MARC, but only through the use of the SMSG syntax in MARC. On the other hand, a command called MSG is available through MARC, but serves a function different from that of the MSG system command. The FILES, MSG, and SMSG commands in MARC are described under “MARC Commands” earlier in this section.

Primitive Commands

Primitive commands are system commands that are processed directly by the master control program (MCP), bypassing the ordinary system command handler. In many cases, you can use primitive commands even when a software failure makes it impossible to use ordinary system commands. All primitive commands begin with two question marks (??).

Three primitive commands are available in MARC. You can use these commands only when MARC is running at an ODT. Two of these commands, *??MARC* (Transfer to Menu-Assisted Resource Control) and *??ODT* (Transfer from Menu-Assisted Resource Control), are used to take the ODT in and out of data comm mode. For descriptions of these commands, see “Configuring the ODT for Data Comm Mode” in Appendix B, “Installing MARC on Your System.”

The other available primitive command is the ??MEMDP (Dump Memory) primitive command. This command has the same syntax and meaning as the ??DP (Dump) primitive command. The only difference is that ??MEMDP is available in data comm mode, whereas ??DP is not. (The ??DP command is unavailable in data comm mode because it could be confused with the ??DP command in CANDE if it were used in a CANDE dialogue.) For more information about primitive commands, see the *A Series System Commands Operations Reference Manual*.

WFL Statements Entered as Commands

You can enter WFL statements as commands in the *Action* field, or on the COMND screen displayed by the COMND action. All WFL statements are available in MARC. Appendix C, “Commands Available in MARC,” lists the WFL statement names and functions. For detailed information about WFL statements, see the *A Series Work Flow Language (WFL) Programming Reference Manual*.

When entered in MARC, all WFL statements except ADD, CHANGE, COPY, REMOVE, RUN, SECURITY, and START must be preceded by the word *WFL*. Preceding a statement with *WFL* causes it to be directed to the WFL compiler.

You can also create a WFL program on the COMND screen by listing a number of WFL statements separated by semicolons (;). The WFL program cannot exceed the boundaries of one COMND screen, which provides 15 lines for command input.

WFL statements that are initiated through MARC follow the same syntax as WFL statements initiated through CANDE. The only exception to this rule is the RUN statement. The RUN statement, when initiated through MARC, rejects variables used to represent task attributes. The task attributes themselves, however, are accepted.

The following WFL statements are available as MARC menu selections:

- ADD
- COPY
- START

The ADD and COPY statements are implemented through the COPY menu selection in MARC. MARC also provides CHANGE, REMOVE, and SECURITY menu selections that serve the same purposes as the WFL statements with the same names without making use of the WFL compiler. MARC rejects the serial-number modifier in the menu-initiated CHANGE and REMOVE commands. Therefore, for operations involving serial numbers, use the *WFL <command>* form rather than the equivalent CHANGE and REMOVE menu selections.

MARC also rejects any CHANGE or REMOVE command that attempts to remove or change a file with the LOCKEDFILE file attribute equal to TRUE. See the *A Series File Attributes Programming Reference Manual* for information on the LOCKEDFILE attribute.

Entering any WFL statement causes MARC to enter tasking mode. This mode is also initiated when any of the WFL statements ADD, COPY, or START is executed by

menu selection. See “Entering Tasking Mode” in Section 4, “Running Programs from MARC,” for more information.

Print System Commands

You can enter any of the PS (Print System) commands in the *Action* field, or on the COMND screen displayed by the COMnd action. Also, the PS menu in MARC provides an alternate way to achieve the functions of many of these commands.

Entering *PS* in the *Action* field displays a list of the PS commands and a brief description of their functions. The PS commands are described in the *A Series System Commands Operations Reference Manual* and in the *A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide*.

Controlling Session Print Attributes

In addition to the PS commands, the following commands are available in MARC to control session print attributes:

Command	Function
DESTNAME	Establishes the station name of a remote printer to be used for output from tasks initiated from the current MARC dialogue
JOBSUMMARY	Establishes whether a job summary is to be printed for a job initiated through MARC
JOBSUMMARYTITLE	Establishes the name of a file to be used for job summary information
NOJOBSUMMARYIO	Establishes whether job summary information is to be produced for tasks initiated from the current MARC dialogue
PRINTDEFAULTS	Establishes a setting for the default printing device for tasks initiated from the current MARC dialogue

The descriptions on the following pages provide more information on these commands.

DESTNAME Command

The DESTNAME command assigns a value to the DESTNAME task attribute.

Syntax

```
— DESTNAME [ = ] [ <station name> ]
```

Explanation

The DESTNAME task attribute specifies the station name of the remote printer to be used for tasks. Once a DESTNAME value is assigned on a particular MARC dialogue, all tasks initiated from that dialogue are printed on the specified remote printer.

Using Commands

The station name must be a valid station name of a remote printer recognizable by COMS, or a warning message is given.

The period (.) option routes all output to the site printer.

Entering *DESTNAME* without succeeding text displays the current DESTNAME value.

The DESTNAME value is established during the log-on procedure by the USERDATAFILE attribute CANDEDESTNAME. See the *A Series Security Administration Guide* for information on how the CANDEDESTNAME attribute value is defined and modified. The DESTNAME value is also inherited from the value for MARC dialogue 1 (MARC/1) when additional MARC dialogues are opened.

JOBSUMMARY Command

The JOBSUMMARY command specifies whether a job summary is printed along with the output from a job.

Syntax

```
— JOBSUMMARY [ = ] [ DEFAULT | CONDITIONAL | SUPPRESSED | UNCONDITIONAL | ABORTONLY ]
```

Explanation

A job summary printout includes such information as the status of a job, the names of programs that run as part of the job, the clock times for each program run as part of the job, the amount of system memory used, and the system messages.

The JOBSUMMARY value is inherited from the value for MARC dialogue 1 (MARC/1) when additional MARC dialogues are opened.

You can assign the following values to JOBSUMMARY:

Value	Effect
DEFAULT	Indicates that no special action is to be taken to override the system-wide options established by the system administrator. DEFAULT is the default JOBSUMMARY value.
CONDITIONAL	Causes job summary printing to occur only if printed or printer backup information is generated by one or more tasks during the session.
SUPPRESSED	Prevents job summary printing whenever print information is generated by tasks during the session.
UNCONDITIONAL	Causes job summary printing to occur unaffected by other conditions that might arise.
ABORTONLY	Causes job summary printing to occur only if the job or any of its tasks terminate abnormally.

The value of the NOJOBSUMMARYIO attribute, discussed later in this section, directly affects the value of the JOBSUMMARY attribute. MARC issues a warning whenever the JOBSUMMARY value is set or is the subject of an inquiry, but only if the NOJOBSUMMARYIO attribute is set to TRUE.

JOBSUMMARYTITLE Command

The JOBSUMMARYTITLE command enables you to create a permanent file to contain the job summary information for a given job.

Syntax

```
— JOBSUMMARYTITLE [ = ] [ <file title> ]
```

<file title>

```
—<file name> [ ON —<family name> ]
```

<file name>

```
[ ( —<usercode>— ) ] /12\ [ <name constant> ]
[ * ] [ # —<string primary> ]
```

Explanation

This command assigns a value to the JOBSUMMARYTITLE task attribute. The system checks the JOBSUMMARYTITLE value at the completion of the job. If no value has been assigned, no job-summary file is created; otherwise, the system creates a permanent file using the value of this attribute as the title of the file.

The file title must follow the standard syntax for a file title. Specifying a file title preceded by a usercode requires privileged-user status.

The period (.) option clears the JOBSUMMARYTITLE value. Entering JOBSUMMARYTITLE without inserting succeeding text causes the current JOBSUMMARYTITLE value to be displayed.

The JOBSUMMARYTITLE value is inherited from the value for MARC dialogue 1 (MARC/1) when additional MARC dialogues are opened.

NOJOBSUMMARYIO Command

The NOJOBSUMMARYIO command establishes a value for or displays the value of the NOJOBSUMMARYIO task attribute.

Syntax

```
— NOJOBSUMMARYIO [ = ] [ TRUE | FALSE ]
```

Explanation

This command sets or displays the value of the NOJOBSUMMARYIO task attribute for tasks run from the MARC dialogue where the NOJOBSUMMARYIO value is set. The system sets the NOJOBSUMMARYIO value for a session job summary based on the previous value for tasks run during the session. Setting the NOJOBSUMMARYIO attribute to TRUE prevents the writing of the job summary information to the job code file, thus saving processing time.

When this attribute is set to TRUE, you can view the job summary information only by using the LOGANALYZER utility to extract the information from the system log file. For more information, see the *A Series System Software Support Reference Manual*.

By default, the NOJOBSUMMARYIO value is FALSE, indicating that normal job summary input and output is to take place for each task. Setting the NOJOBSUMMARYIO value to TRUE might affect the performance of the JOBSUMMARY attribute.

The NOJOBSUMMARYIO value is inherited from the value for MARC dialogue 1 (MARC/1) when additional MARC dialogues are opened.

PRINTDEFAULTS Command

The PRINTDEFAULTS command designates the default printing device for the current MARC dialogue.

Syntax

```
— PRINTDEFAULTS [ = ] [ <print defaults> ]
```

Explanation

The PRINTDEFAULTS command assigns a value to the PRINTDEFAULTS task attribute. The PRINTDEFAULTS value that is assigned on a particular MARC dialogue is used for all tasks run from that dialogue.

For information on the print defaults construct, see the *A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide* under the task attribute PRINTDEFAULTS. The PRINTDEFAULTS attribute is solely a task attribute and is unrelated to a session. Using the period (.) option clears the current value. Entering *PRINTDEFAULTS* without succeeding text displays the current PRINTDEFAULTS value.

MARC rejects any attempt to set the PRINTDEFAULTS value unless MARC is linked to the PRINTSUPPORT library. MARC uses the validation function of the PRINTSUPPORT library to monitor the syntax of the print defaults construct, and notes specific errors in the syntax when it finds them.

The PRINTDEFAULTS value is inherited from the value for MARC dialogue 1 (MARC/1) when additional MARC dialogues are opened.

System Utilities

Many menu selections in MARC initiate system utilities. You can also access these utilities through MARC by entering a command of the following form in the *Action* field:

```
RUN *SYSTEM/<utility name>
```

When you use MARC to run a system utility, MARC initiates tasking mode. For more information about this mode, see “Entering Tasking Mode” in Section 4, “Running Programs from MARC.”

The remainder of this section lists the utilities that are available through MARC. The list is separated into groups of related utilities.

System Administrator Utilities

The system administrator can use these utilities to configure and manage the system. The following list shows the utilities included, along with the title of the documentation that describes each of them. In some cases, you are referred to online documentation in MARC. In these cases, enter the designated HELP command in the *Action* field or the *Choice* field of the screen indicated.

Utility	Documentation
BARS	<i>A Series System Software Support Reference Manual</i>
COMS Utility	<i>A Series Communications Management System (COMS) Configuration Guide</i>
CONFIGURATOR	<i>A Series System Configuration Guide</i>
HARDCOPY	<i>A Series System Software Support Reference Manual</i>
INSTALL/BNA	<i>A Series BNA Version 1 Operations Guide</i>
Interactive Configurator	<i>A Series System Configuration Guide</i>
Interactive Datacomm Configurator (IDC)	<i>A Series Interactive Datacomm Configurator (IDC) Operations Guide</i>
LISTVOLUMELIB	HELP LIST on the UTIL menu
LOGANALYZER	<i>A Series System Software Support Reference Manual</i>
LTTABLEGEN	<i>A Series Printing Utilities Operations Guide</i>

continued

Using Commands

continued

Utility	Documentation
MAKEUSER	<i>A Series Security Administration Guide</i>
PATCHCONTROLWARE	HELP CON on the UTIL menu
PRINTCOPY	<i>A Series System Software Support Reference Manual</i>
RLTABLEGEN	<i>A Series System Software Utilities Operations Reference Manual</i>

Debugging Utilities

These utilities are used to help analyze and correct problems on the system. The following list shows the utilities included, along with the title of the documentation that describes each of them. In some cases, you are referred to online documentation in MARC. In these cases, enter the designated HELP command in the *Action* field or the *Choice* field of the screen indicated.

Utility	Documentation
DCAUDITOR	<i>A Series System Software Support Reference Manual</i>
DCDLPDUMPANALYZER	HELP DCDLPD on the UTIL menu (Micro A through A 6 systems only)
DCSTATUS	<i>A Series System Software Support Reference Manual</i>
DUMPALL	<i>A Series System Software Utilities Operations Reference Manual</i>
DUMPANALYZER	<i>A Series System Software Support Reference Manual</i>
LOGANALYZER	<i>A Series System Software Support Reference Manual</i>
NDLIANALYZER	<i>A Series Network Definition Language II (NDLII) Programming Reference Manual</i>
NSPDUMPANALYZER	HELP NSPD on the UTIL menu
PATCHCONTROLWARE	HELP CON on the UTIL menu

General Utilities

These utilities are of general use to all users of the system. The following list shows the utilities included, along with the title of the documentation that describes each of them. Some utilities might be unavailable to you. For example, if your site does not use InfoExec™ software, you cannot access InfoExec utilities.

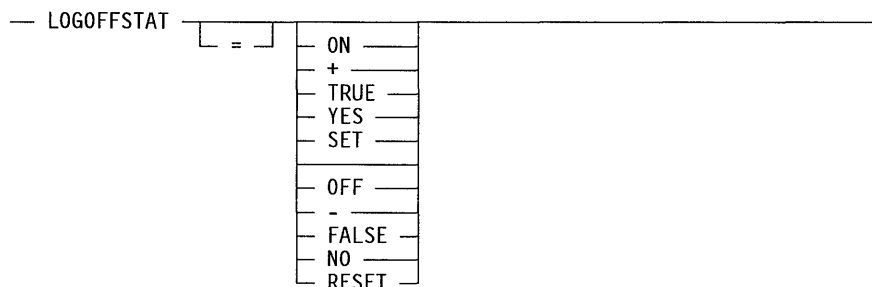
Utility	Documentation
Advanced Data Dictionary System (ADDS)	<i>A Series InfoExec Advanced Data Dictionary System (ADDS) Operations Guide</i>
BACKUP and Backup Processor	<i>A Series Printing Utilities Operations Guide</i>
CARDLINE	<i>A Series System Software Utilities Operations Reference Manual</i>
COMPARE	<i>A Series System Software Utilities Operations Reference Manual</i>
Extended Retrieval with Graphic Output (ERGO)	<i>A Series Extended Retrieval with Graphic Output (ERGO) Operations Guide</i>
FILECOPY	<i>A Series System Software Utilities Operations Reference Manual</i>
FILEDATA	<i>A Series System Software Utilities Operations Reference Manual</i>
GUARDFILE	<i>A Series Security Features Operations and Programming Guide</i>
HELP	<i>A Series Help Utility Operations Guide</i>
InfoExec	<i>A Series Data Management Functional Overview</i>
Intelligent Laser Printing System (ILPS) Forms Manager	<i>A Series Intelligent Laser Printing System (ILPS) Forms Manager Operations Guide</i>
Interactive DataComm Configurator (IDC)	<i>A Series Interactive Datacomm Configurator (IDC) Operations Guide</i>
Interactive Menugraph Generator (IMG)	<i>A Series Interactive Menugraph Generator (IMG) Operations Guide</i>
INTERACTIVEXREF	<i>A Series System Software Utilities Operations Reference Manual</i>
Message Translation Utility (MSGTRANS)	<i>A Series Message Translation Utility (MSGTRANS) Operations Guide</i>
Operations Control Manager (OCM)	<i>A Series DMSII Utilities Operations Guide</i>
PATCH	<i>A Series System Software Utilities Operations Reference Manual</i>
Screen Design Facility (SDF)	<i>A Series Screen Design Facility (SDF) Operations and Programming Guide</i>
Screen Design Facility Plus (SDF Plus)	<i>A Series Screen Design Facility Plus (SDF Plus) Installation and Operations Guide</i>

Using the LOGOFFSTAT Option

The MARC option LOGOFFSTAT controls the display of session resource information when the HELLO, SPLIT, or BYE commands are used. You must have SYSTEMUSER privileges to use this option.

You can use the LOGOFFSTAT option from the *Action* field of the MARC home menu or from the LOGOFFSTAT screen.

Syntax



Explanation

If you enter only *LOGOFFSTAT*, the current setting of the option is displayed.

If you set the LOGOFFSTAT option with the value ON, +, TRUE, YES, or SET, the following information appears when you enter a HELLO, SPLIT, or BYE command:

- The date and time when the session was terminated
- The usercode that started the session
- The duration of the session
- The processor time used by the current user from within MARC
- The I/O time used by the current user from within MARC

If you enter the STATUS or WRU command, the processor time and the I/O time are also included in the information displayed. These times include only time spent within MARC; they do not include time used in other COMS windows such as for applications, CANDE, or programs run through MARC.

If you enter the LOGOFFSTAT option using the value OFF, -, FALSE, NO, or RESET, only the following information appears when you enter a HELLO or SPLIT command:

- The date and time when the session was terminated
- The usercode that started the session
- The duration of the session

When the LOGOFFSTAT option is reset, no information is displayed with the BYE command; the blank LOGON screen appears instead.

By default, this option is reset.

Figure 6-2 illustrates the LOGOFFSTAT screen. To set the option with this screen, enter an X in the appropriate box.

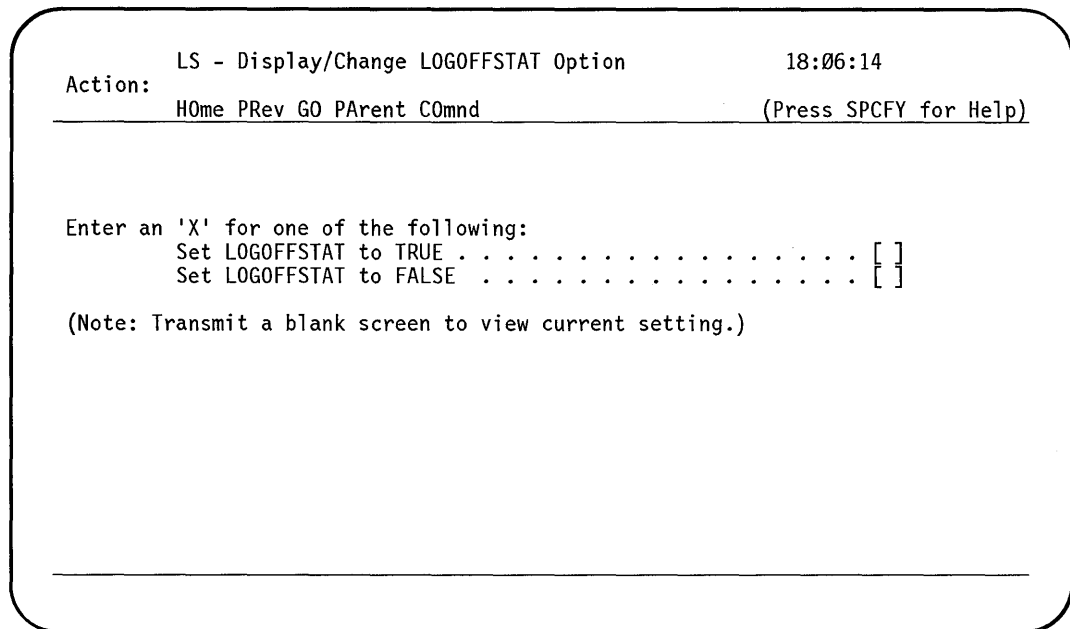


Figure 6-2. Logoff Statistic Screen

Special Considerations

The session statistics appear in the normal message area reserved for each command. However, if the automatic log-on feature is set for the user, the log-on or log-off statistics appear as a single-line-mode message. This display mode is also used when the host is disconnected from COMS, usually for one of the following reasons:

- A COMS station timeout event occurs.
- The user had station-transferred to the host running MARC, and has been disconnected—for example, by entering a BYE request.
- The station is a network support processor (NSP) switched-line station and has been disconnected.
- The station has set the CP 2000 terminal option LOGOFFDISCONNECT and has been disconnected.

Examples

The following examples illustrate selected uses of the LOGOFFSTAT option.

Using Commands

- The following example lists the current setting for the LOGOFFSTAT option:

```
LOGOFFSTAT
```

The Logoff-Statistics option is turned OFF.

- The following example sets the LOGOFFSTAT option and shows the statistics displayed when the session is terminated using the SPLIT command:

```
LOGOFFSTAT = ON
```

```
Session 7009 ended at 13:40:23 on 9/19/91. Duration = 5:13:50.3;  
Proc time = 4:00:1.25; I/O time = 3:30:0.18. New session 7301; User= xx.
```

Section 7

Controlling System Security

The security features available through MARC are the same as those available through CANDE, and are documented in the *A Series Security Features Operations and Programming Guide*. This guide also provides a detailed description of the access rights associated with usercodes.

To have access to MARC, you must log on under a usercode, which normally has an associated password and might have an associated accesscode, chargecode, or both. For more information, see “Logging On” in Section 2, “Starting to Use MARC.”

MARC suppresses from the menu displays any selections initiating system functions that your usercode does not authorize you to use. Also any attempt to use commands that exceed your privileges while entering commands directly results in an error.

The usercode that you use to log on to the MARC/1 dialogue is inherited by any further MARC dialogues you open. The security status of that usercode is also inherited by each of the dialogues. That is, the same privileges available at dialogue 1 are available at each of the other MARC dialogues. See Section 5, “Using COMS Windows and Dialogues,” for more information on the MARC dialogues.

Understanding Security Categories

A number of A Series security categories affect MARC operation. Some of these categories are associated with individual usercodes, while others apply to groups of users or individual stations.

Usercode Security Categories

The security categories associated with the usercode that you use to log on determine the system functions you are permitted to access. Your security administrator defines your usercode and its various attributes in the USERDATAFILE usercode database. The following paragraphs describe the security categories your security administrator can associate with your usercode.

Security Administrator

The security administrator category is effective only on systems with the optional InfoGuard security enhancements.

When the value of the system option SECADMIN is TRUE, only a usercode with security-administrator status has access to MARC menus and functions that control security on the system. For example, only a security-administrator usercode can access

Controlling System Security

the MARC security options menu, or run the Simple Installation (SI) program that installs system software.

When the value of the system option SECADMIN is TRUE, only a usercode with security-administrator status can create or modify the USERDATAFILE usercode database. If the SECADMIN value is FALSE, or if your system has no security-administrator usercodes defined, any privileged usercode can create or modify the USERDATAFILE usercode database.

See the *A Series Security Administration Guide* for details on the role of a security administrator.

SYSTEMUSER

A usercode that is marked as a SYSTEMUSER is permitted access to MARC menus and selections that affect the operation of the system. In addition, a SYSTEMUSER usercode enables you to enter in the *Action* field all system commands – except primitive commands – and BNA Version 1 or Version 2 network commands.

Some system functions are available to all usercodes, even those without SYSTEMUSER status. These functions include the use of Work Flow Language (WFL) statements, system utilities, and menu selections that display information about the system. MARC accepts the RESTART (Restart Jobs) system command from users without SYSTEMUSER status.

Control Capable

A control-capable usercode is permitted access to MARC menus and selections that control the operation of COMS. In addition, your site can designate that control-capable usercodes have access to the COMS Utility, which defines and maintains the contents of the COMS configuration file. A usercode with control-capable status also enables you to enter COMS commands in the *Action* field.

Control-capable privileges are also available if you log on under any usercode at a terminal that has been designated in COMS as having control-capable status.

Privileged User

A usercode that is marked as privileged enables its user to access all system and user files, including files under other usercodes. Privileged users can also use operating system procedures such as GETSTATUS and SETSTATUS, which display or set the status of jobs, tasks, and peripherals, as well as of the system configuration and the operating system itself.

If the value of the system option SECADMIN is TRUE, but no usercode is designated as a security administrator in the USERDATAFILE usercode database, any privileged user can run the MAKEUSER utility in MARC, either from the USER or UTIL menus, or from any *Action* field.

Conferring SYSTEMUSER status on a usercode does not make the usercode privileged, although it is possible for a usercode to be both privileged and a SYSTEMUSER.

Other Security Categories

The following security categories are not associated with individual usercodes but affect MARC in other ways.

Super User

The system administrator can use the COMS Utility to define one or more terminals as *super-user-capable stations*. At a super-user-capable station, any user can log on to MARC by specifying an asterisk (*) in the usercode field of the LOGON screen, instead of supplying a usercode and password. The asterisk (*) usercode has SYSTEMUSER status, but is not privileged.

In other words, *super user* is a usercode security category like privileged or SYSTEMUSER, except that super-user status is not associated with an individual usercode. Instead, the status is associated with a particular station. A usercode has super-user status only when the following conditions are both true:

- The user accesses the system from a super-user-capable station.
- The user enters an asterisk (*) in the usercode field during the log-on sequence.

When a user logs on as a super user, the MARC home menu appears. The normal features of MARC are available to the user, but if the user opens a dialogue with CANDE, he or she must at that point log on to CANDE in the conventional fashion, with a usercode and (if required) password.

System commands entered by a super user are given the same privilege as system commands entered from an ODT. However, the system rejects primitive commands – those preceded by two question marks (??) – from a super user.

The actions of a user who is not logged on under a usercode might be difficult to monitor. Most installations therefore limit the COMS super-user feature to stations with a high degree of physical security. If security is a concern on your system, this feature might be unavailable.

COMMANDCAPABLE

COMMANDCAPABLE is a COMS security category that permits security restrictions to be imposed upon selected users.

If the system administrator uses the COMS Utility to define the COMMANDCAPABLE security category for the system, usercodes without COMMANDCAPABLE specified in their security category list have special capabilities and are subject to special limitations, as follows:

Controlling System Security

- After logging on under these usercodes, users are limited to the following commands:
 - BYE
 - CLOSE
 - END
 - ON
 - PASSWORD
 - SHOW PASSWORD
 - GENERATE PASSWORD
 - PASS
 - PURGE
 - RESUME
 - SUSPEND
 - WINDOWS
 - WRU
- The security administrator can establish the FUNCTIONS menu as the MARC home menu for these users. The security administrator makes this specification through the USERDATAFILE usercode database, as described under “Setting the Usercode Default Home Menu” in Section 8, “Creating Custom Versions of MARC.” From the FUNCTIONS menu, these users can view introductory information about MARC, use the commands just listed, either directly or through the associated forms, and log off.
- These users can activate MARC screens, but are limited to the commands just listed.
- In a MARC menugraph, the ON command is available to these users to provide access other COMS windows. Thus, a user can be restricted from general access to the system, yet still have access to specific windows dedicated to some unique function.

See the *A Series Communications Management System (COMS) Configuration Guide* for information on the COMS Utility. For instructions on how to establish COMMANDCAPABLE as a valid security category and how to assign its capability to users, see the *A Series Security Administration Guide*.

If the COMMANDCAPABLE security category is added or deleted, the system operator must reinitialize MARC and COMS to activate the change.

Restricting Access to MARC Screens

The system administrator can restrict access to MARC screens by using the Interactive Menugraph Generator (IMG) to remove the GO action from the action line in the MARC menugraph. For information about IMG, see the *A Series Interactive Menugraph Generator (IMG) Operations Guide*.

When the *GO* action has been removed, it becomes a valid command only for privileged and *SYSTEMUSER* usercodes, or super users. These statuses are discussed earlier in this section. *MARC* rejects any attempt to enter *GO* that is made by a user who is neither privileged, nor identified as a *SYSTEMUSER*, nor logged on as a super user. *MARC* responds to such attempts with an error message.

This blocking feature applies only to the original *MARC* screens provided by Unisys; it does not apply to user-created screens. That is, if you create a screen that includes a *GO* action, you cannot limit that action to users with privileged, *SYSTEMUSER*, or super-user status by removing that action from the action line in the menugraph as you can on an original *MARC* screen.

Setting the Security Privileges

The privileges of a given usercode are recorded in the *USERDATAFILE* usercode database on the system. *MARC* assigns to your usercode whatever privileges are specified for it in the *USERDATAFILE* unless *COMS* has been set to limit the privileges that are available at your station.

Using *COMS* to Set Privileges

The *COMS* Utility is the tool used for setting the privileges associated with a particular *COMS* station – that is, a terminal under the control of *COMS*. The *COMS* Utility is included only in the *COMS* (Full-Featured) message control system (*MCS*) – that is, it is not included in *COMS* (Kernel).

A request to execute the *COMS* Utility is accepted only if it comes from one of the following sources:

- An *ODT* running in data comm mode
- A remote station marked as a *COMS* control station
- A usercode that has control capability

If no control stations are defined in the Network Definition Language II (*NDLII*) when the system is initialized, the *COMS* Utility is available only through the *ODT* until further control stations are defined.

A remote terminal becomes a super-user-capable station when it is designated as such by the *COMS* Utility. An *ODT* is automatically considered a super-user-capable station. *COMS* (Full-Featured) provides the *COMS* Utility window, which enables you to designate remote stations as super-user-capable stations as well. *COMS* (Kernel) grants super-user status only on *ODT* stations.

Changing the *USERDATAFILE* Usercode Database

The primary tool for modifying the *USERDATAFILE* usercode database is the *MAKEUSER* utility. If security-administrator status has been enabled, only a security

administrator can access MAKEUSER. If no security administrator is defined, only a user running under a privileged usercode is permitted to access the MAKEUSER utility.

The only exception to this rule occurs when no usercodes have yet been declared on the system. In this situation, anyone at an ODT or a remote super-user-capable station can log on using an asterisk (*) to run the MAKEUSER utility and create usercodes for the system. See the *A Series Security Administration Guide* for more information on the MAKEUSER utility and the USERDATAFILE usercode database.

Changing the Password for a Usercode or Accesscode

While running MARC, you can use the USER menu to change the password associated with your usercode or accesscode. Alternatively, you can use the PASSWORD command in COMS. You cannot make chargecode changes through MARC.

See the *A Series Communications Management System (COMS) Operations Guide* for more information on the PASSWORD command in COMS.

Writing Local Security Procedures

You can implement security procedures that are local to your installation.

SECURITYSUPPORT Library

To implement local security procedures, you write a program that MARC can call on to perform additional security checks. You must perform the following steps concerning this program:

- Mark the program as *privileged transparent* by using the PP (Privileged Program) system command in the form PP:T.
- Declare the program as the SECURITYSUPPORT library by using the SL (Support Library) system command.

The PP and SL commands are both available through MARC and are described in the *A Series System Commands Operations Reference Manual*.

Excessive Log-on Attempts

You might want to address how your system responds to a user who makes repeated unsuccessful attempts to log on to the system.

Understanding the Default Procedure

By default, a user at a CP 2000 station or data comm terminal is limited to 10 consecutive invalid attempts to log on. Immediately after the 10th consecutive invalid attempt, the system rejects all usercodes and passwords entered from that station or terminal. MARC reports the number of failed log-on attempts to the SECURITYSUPPORT library and any monitor stations on the system.

To determine if this process is in effect for a particular station, enter the COMS command `STATUS STATION <station ID>`. If the system is rejecting usercodes, the following message is displayed:

```
Logon is prohibited until station is made READY
```

A station in this state must be made ready before the system again validates log-on attempts from that station. To resume usercode/password validation, use one of the following commands:

```
READY <station name>  
READY <LSN>
```

Examples

The following examples illustrate selected uses of the READY command.

- The following command resumes usercode/password validation for the station with the station name TA243:

```
READY TA243
```

- The following command resumes usercode/password validation for the station with the logical station number 415:

```
READY 415
```

Modifying the Maximum Number of Log-on Attempts

The system command SECOPT (Security Options) enables you to specify a new value for the MAXLOGONRETRIES option, the option that defines the number of attempts a user is permitted to make before the station becomes unready. You can specify a value between 0 and 15. A value of 0 implies an unlimited number of attempts. The default value is 10.

When the value of the system option SECADMIN is TRUE, only a usercode with security-administrator status can use the SECOPT command. When the SECADMIN value is FALSE, or if your system has no security-administrator usercodes defined, PU (privileged user) status is sufficient. See the *A Series Security Administration Guide* for information on this command.

Accessing Security and Help Books

Help book files can have a SECURITYTYPE value of PRIVATE or PUBLIC. In MARC, you can access both types of books.

The Help Utility creates help books with a SECURITYTYPE value of PRIVATE. MARC ensures that only authorized users can access books that are marked PRIVATE. To permit access by all users, change the security of the MARC help books to PUBLIC. You can change the SECURITYTYPE value of a help book in either of the following ways:

- Access the FILE menu, available from the MARC home menu, and choose the SEC selection.
- Execute a SECURITY command in CANDE. See the *A Series CANDE Operations Reference Manual* for information on the SECURITY command.

Section 8

Creating Custom Versions of MARC

MARC provides several features that enable you to customize the system for the requirements of your site. You can

- Create custom versions of the MARC menugraph.
- Use multiple MARC menugraphs on your system.
- Create custom versions of the MARC help text.
- Create additional help text.
- Write COMS processing items for MARC.
- Create directive commands.
- Relocate internal MARC files to improve performance.

This section describes these customizing features. In addition, you can set and change options that control the language in which MARC screens and help text are displayed on your system. See Section 9, “Using MARC Internationally,” for more information on setting and changing the display language.

Customizing the MARC Menugraph

The primary tool you use to customize the MARC menugraph is the Interactive Menugraph Generator (IMG). See the *A Series Interactive Menugraph Generator (IMG) Operations Guide* for instructions on creating customized menugraphs.

You can use IMG to change the contents of the standard MARC menus and forms, and to add new menus and forms. You can also designate a menu other than the standard MARC home menu as the home menu for a particular user. The newly designated menu can be an existing MARC menu or a new menu that you create for one or more users.

This capability enables you to create, for example, a customized home menu for users in the accounting department of your installation, and another customized home menu for the sales department. Users in other departments of the installation could use the standard MARC home menu.

The MARC menugraph includes several forms, known as *internal forms*, that have no commands linked to them. The system menugraph file must include all the internal forms at all times. If the internal form named ALEXIS is missing from the system menugraph file, the system cannot display a LOGON screen.

If any of the other internal forms are missing, MARC displays warning messages and waits for further instructions.

Creating Custom Versions of MARC

If internal forms are missing, take one of the following courses of action:

- Replace the system menugraph file with one that has all the internal forms in it, and then enter *AX* as your response to the warning message.
- Enter *AX OK* to start MARC with the current incomplete menugraph file.

Note: *Be aware that starting MARC with an incomplete menugraph file can cause MARC to fail with the message INVALID INDEX when it attempts to display a screen based on a missing internal form.*

Once you have created the customized MARC menugraphs, you can upgrade those menugraphs to make them compatible with subsequently installed versions of MARC. To perform such upgrades, use the BMENU utility. See the *A Series Interactive Menugraph Generator (IMG) Operations Guide* for information on using BMENU.

Changing the Screens

You can add to, delete, or change any of the MARC menus and forms to meet the particular needs of your installation. IMG provides the means to make the majority of these alterations.

IMG leads you through the screen-editing process by presenting you with a series of selections that enable you to build screens piece by piece. Menus and forms are stored in a file known as a *menugraph file*. You can use IMG either to customize the supplied menugraph file or to create new menugraph files.

Portions of some screens are stored as messages. You can change these portions by using the Message Translation Utility (MSGTRANS). Section 9, “Using MARC Internationally,” outlines the process for translating messages into another language; see “Localizing MARC Messages.” For detailed information on MSGTRANS, see the *A Series Message Translation Utility (MSGTRANS) Operations Guide*.

Customizing Menus

IMG enables you to change the selection key or the selection description associated with a menu selection. You can also change the short help or long help keyword and the help book name associated with a menu selection. In addition, you can change the command generated by a menu selection. For example, a site might need to change the RUN statement for certain utilities to specify a different family name.

You can use IMG to delete selections or add new selections to a menu. You can create entirely new menus and add them to the menugraph file. For example, you can create a home menu that contains selections to run the most frequently used applications at your site. Using IMG, you can designate this new menu as the default home menu. Doing so ensures that the new menu is the first one to be displayed after a user logs on. Such a menu should include a selection to display the home menu supplied by Unisys (the one named *MARC*). Associate the command *GO MARC* with that selection.

Any new menu selections that you add must invoke either commands that are already available through MARC or screen-traversal commands that cause another screen to be

displayed. See Appendix C, “Commands Available in MARC,” for more details. See the *A Series Interactive Menugraph Generator (IMG) Operations Guide* for information on screen-traversal commands.

Customizing Forms

Forms are less modifiable than menus. You can use IMG to change any of the following components of a form:

- The description text on a form
- The default value for each field
- The short help and long help keywords and the help book name associated with each field
- The command associated with a form

You cannot change the position of the input fields on a form. You can create new forms, but they must follow the format of one of the general form templates provided by IMG. A general form template has a fixed number of input fields and can be adapted to initiate any command that is available through MARC. However, if the command requires more parameters than are provided on any of the form templates, preset values must be used for the extra parameters.

If general, specific, or internal forms are present in a menugraph, MARC checks the version number of the format to be sure that the format version matches the version of MARC. A failure in the version check is handled as if the screen cannot be found. See the *A Series Interactive Menugraph Generator (IMG) Operations Guide* for detailed instructions on menugraph generation.

Using the Message Translation Utility (MSGTRANS)

Certain portions of each MARC screen are determined by the MARC messages that are currently in use rather than by the menugraph. You can change these portions of the screen by using the Message Translation Utility (MSGTRANS). See “Localizing MARC Messages” in Section 9, “Using MARC Internationally,” for a list of these portions of the screen. See the *A Series Message Translation Utility (MSGTRANS) Operations Guide* for detailed instructions on using the MSGTRANS utility.

Setting the Usercode Default Home Menu

For instructions on creating a custom home menu for the entire system, see “Customizing Menus” earlier in this section. If you have access to the MAKEUSER utility, you can also designate a custom home menu for one or more individual usercodes on the system. Doing so enables you to control the first menu that each user sees after logging on. This feature is available through the HOMEMENU usercode attribute of the MAKEUSER utility, as described in the *A Series Security Administration Guide*.

Creating Custom Versions of MARC

A home menu defined through the MAKEUSER utility is known as the *usercode default home menu*. This menu can be any menu in the standard MARC menugraph, such as the FUNCTIONS menu, or it can be a custom menu created for a particular user.

Because any menu can be declared to be the home menu, the home menu is not necessarily located at the root of the menu structure. The home menu can have a parent screen that is accessible by way of the PARENT action on a MARC screen.

You can always reach the screen that is at the root of the menu structure by using the PARENT action repeatedly. The screen that is at the root of the menu structure is used as the default home menu if there is no HOMEMENU attribute specification for a usercode in the USERDATAFILE usercode database, or if the specified home menu cannot be found in the menugraph.

Making a Custom Menugraph Available to Users

Suppose that you have used IMG, MSGTRANS, or both to customize an existing menugraph file and you want all users of the old version to use the new version. If no MARC users are currently accessing the old version of the menugraph, you need only delete the old version of the file and give the new version the same title as the old version.

Caution

Deleting files that MARC is currently running or has used since it was started can severely degrade system performance.

If MARC users are currently accessing the old version of the menugraph, you can make the new menugraph immediately available by entering the NEW command with the MENU option.

Specifically, perform the following steps:

1. To determine if any users are currently accessing the old version, enter a command of the form *FILE <menugraph file title>* in the *Action* field. For example:

```
FILE *SYSTEM/MARC/MENUGRAPH/ENGLISH ON DISK
```

If the first line of the resulting display contains the words *IN USE*, at least one user is currently accessing the menugraph.

2. In this case you can still make the new menugraph available immediately by entering a command in the form *NEW MENU <menugraph file title>*.

MARC searches for the specified menugraph file on the disk family where the file MARC/SYSTEM/COMMANDER resides. This file is the main body of MARC.

If MARC fails to find the menugraph file on that family, MARC searches the substitute family that is specified for the current dialogue.

If the menugraph file is on another family, include an *ON <family name>* clause in the menugraph file title when you use the NEW command. When you include this specification, MARC searches for the menugraph file only on the specified family.

For example:

```
NEW MENU (ADMIN)CUSTOM/MARC/MENUGRAPH/ENGLISH ON CUSTFILES
```

The NEW MENU command makes an updated version of an in-use menugraph file available for access by users without terminating MARC. The NEW MENU command causes MARC to search for the menugraph file with the given title and to refresh its internal tables for that menugraph file. In doing so, MARC closes the physical file, permitting a new menugraph file with the same title to be accessed.

Leftover Screens

Even after a new menugraph version is made available, some screens might temporarily be presented in their versions from the old menugraph. This situation arises because MARC stores the last 15 screens that were displayed in the current dialogue. These screens are stored in a system memory structure called the *screen cache*. The screen actions PRev and RReturn present screens from the screen cache in their original format until they are rolled out of the cache. Forms displayed from the cache include any input field values previously entered on the form.

If the menugraph was updated recently, some screens from the old menugraph version are still in the screen cache and displayed. To ensure that all screens displayed are from the new version of the menugraph, request that users either move to a new MARC dialogue or end their current sessions and log on again.

Who May Update a Menugraph

In general, the SECURITYTYPE attribute of a menugraph file determines whether a nonprivileged user can update that menugraph file. However, menugraphs that are not usercoded can be updated only by a privileged user, or by any user at a super-user-capable station or an ODT. See Section 7, "Controlling System Security," for more information on security restrictions.

Error Messages

The following error messages might appear when you use the NEW MENU command:

- The following warning appears if a file exists with the correct title and its FILEKIND file attribute value is DATA, but the file fails the opening sequence for a menugraph file:

```
BAD <menugraph file title> FILE
```

Creating Custom Versions of MARC

In this case, the NEW command still closes the old version of the menugraph file so that it is no longer accessible. MARC then rejects attempts to open the new menugraph file every time it needs to access a screen in the specified menugraph. Performance can be degraded severely until a valid menugraph file is loaded.

- The following message means that MARC is unable to find a file with FILEKIND = DATA with the specified title on the appropriate family:

```
NO FILE <menugraph file title> FOUND
```

You might have misspelled the menugraph title, or the menugraph file might be located on a different family.

- The following error message means the menugraph file is not in an OPEN state:

```
WAS NOT OPEN
```

Because no MARC users are currently accessing the menugraph, you can remove the old version of the file and give the new version the same title rather than executing a NEW command to update the internal MARC tables for that file.

Using Multiple MARC Menugraphs

Your site can make up to five different MARC menugraphs available to a given user. A maximum of 20 MARC menugraphs can be in use on the system at one time. For example, a site might have the standard MARC menugraph supplied by Unisys for most users, a simplified menugraph for users new to the system, and a menugraph in a foreign language for users who are nationals of other countries. See Section 9, "Using MARC Internationally," for information on menugraphs in foreign languages.

The following paragraphs describe how to control the various menugraphs in use on the system.

Active Menugraph List

The set of menugraphs in use on a system is known as the *active menugraph list*. The complete list of menugraphs is as follows:

Menugraph Name	Explanation
Substitute menugraph	Menugraph searched if a user explicitly requests such a search by using the command <i>GO <screen> IN <menugraph></i>
Usercode default menugraph	Menugraph associated with a particular usercode in the USERDATAFILE
MARC session language menugraph	Menugraph associated with the current MARC session language
System default menugraph	Menugraph associated with the current system language
English language menugraph	Menugraph with the file name SYSTEM/MARC/MENUGRAPH/ENGLISH

A given session might use only some of these menugraphs. To list the menugraphs in use for your session, enter *MENU* in the *Action* field. This command is described later in this section; refer to “Displaying the List of Active Menugraphs.”

The preceding list also shows the order in which MARC searches the menugraph files to find a screen specified by the screen action *GO* <screen name> or *PArent*, or specified by a particular menu selection. MARC searches through these menugraphs and displays the first screen it finds with the appropriate screen name.

If the screen found is a form, MARC determines whether this specific form is currently present in the screen cache. If the form is in the cache, MARC displays the form complete with the input field values previously entered.

Because each MARC dialogue runs a separate MARC session, a different active menugraph list can be in use in each of the MARC dialogues. Only the substitute and MARC session language menugraphs are likely to be different, however, as they can be changed during a session. The other menugraphs in the list would differ from one dialogue to another only if changes were made to the system language, or to certain of the default settings associated with a usercode, between the time a user logged on to MARC and the time the user opened a particular MARC dialogue.

The following paragraphs describe MARC requirements for selection of each menugraph in the active menugraph list.

Substitute Menugraph

A substitute menugraph is a menugraph that MARC searches before any other menugraph for a specified screen. The substitute menugraph remains in effect only for the current dialogue in the current session and only for the usercode under which it is established.

There is no substitute menugraph when you first start a MARC session. You establish a substitute menugraph by entering the following action:

```
GO <screen name> IN <menugraph file title>
```

This action is described later in this section; see “Declaring a Substitute Menugraph.” Once you establish a substitute menugraph, it becomes the first menugraph that MARC searches whenever it is searching for a screen and remains so until one of the following events occurs:

- A different substitute menugraph is established.
- You switch to another MARC dialogue.
- You end the current MARC session.

Usercode Default Menugraph

A usercode default menugraph differs from a substitute menugraph in that the usercode default menugraph, if defined, remains in effect from one MARC session to

another. Users with access to the MAKEUSER utility can define a usercode default menugraph by means of the MENUFILENAME usercode attribute of the MAKEUSER utility, as described in the *A Series Security Administration Guide*.

If the menugraph file to be defined is stored under a usercode—as opposed to an asterisk (*)—that usercode must be included in the MENUFILENAME specification. The usercode default menugraph must reside in the same family as the SYSTEM/MARC/COMMANDER code file.

If your usercode has no default menugraph defined through the MAKEUSER utility, your usercode default menugraph is the same as the system default menugraph.

MARC Session Language Menugraph

The MARC session language menugraph is the menugraph file associated with your current MARC session language. The *MARC session language* is the language you are using for the display of screens and system messages in the current MARC dialogue of the current session. This language is the same as the system language unless you explicitly designate otherwise. See “Setting the MARC Session Language” in Section 9, “Using MARC Internationally,” for more information.

If you designate a MARC session language, MARC looks for messages translated into that language. For information on making messages available in different languages, see “Choosing a Language” in Section 9 and the *A Series Message Translation Utility (MSGTRANS) Operations Guide*.

If no messages are available in the requested language, no MARC session language menugraph is established.

If messages are available in the requested language, MARC then looks for a menugraph file that has a title of the form

```
SYSTEM/MARC/MENUGRAPH/<language name>
```

The language name is the name that was specified when the language was designated, and is a string of up to 17 alphanumeric characters. If MARC finds a file with a title in this form for the requested language, that file becomes the session language menugraph. Otherwise, the system default language menugraph is accepted as the correct menugraph for the newly designated MARC session language.

Note: *To revise the association of the new language with the default system language menugraph file, you must reinitialize MARC.*

System Default Menugraph

The system default menugraph is the menugraph file associated with the system language. The *system language* is the default language for the display of screens and messages throughout the system. The system administrator can establish this language; see “Setting the System Language” in Section 9, “Using MARC Internationally,” for more information. If no other system language is established for a site, the system language is English.

If a language other than English is established as the system language at the time MARC is initializing, MARC looks for messages translated into that language. If no messages are available in the requested language, the MARC session language is set to English, the English menugraph is established, and MARC produces messages and menus in English.

If messages are available in the requested language, MARC then looks for a menugraph file that has a title of the form

```
SYSTEM/MARC/MENUGRAPH/<language name>
```

If MARC finds a file with a title in this form for the requested language, that file becomes the system language menugraph. Otherwise, MARC suspends operation until the user provides appropriate responses to RSVP prompts.

English Language Menugraph

The English language menugraph is the menugraph that the system searches if a requested screen occurs in none of the other menugraphs mentioned previously, and if the English language menugraph is different from all the other menugraphs. The English language menugraph file has the title SYSTEM/MARC/MENUGRAPH/ENGLISH.

Displaying the List of Active Menugraphs

You can use the MENU command to display the names of the active menugraphs and to identify any menugraphs that are absent. MARC displays a list of the menugraphs in the same order that it searches them for screens.

To use this command, enter *MENU* in the *Action* field of any screen that accepts commands.

If MARC cannot find a menugraph file that it accessed earlier in the session, it displays the following message after the file name:

```
WARNING! MARC cannot find this menugraph file.
```

If MARC lists a menugraph file title without displaying this message on the next line, that menugraph file is currently present. However, a menugraph file that is present can still be invalid if it is in a format that MARC cannot use.

Compatibility problems can occur if MARC encounters a menugraph created under a different software release level. MARC can work only with compatible-version menugraphs. If MARC encounters an incompatible menugraph, it displays the following message:

```
WARNING! An invalid menugraph has been encountered.
```

See the *A Series Interactive Menugraph Generator (IMG) Operations Guide* for information on how to deal with compatibility problems concerning menugraphs.

Caution

System performance can be severely impaired if any of the listed menugraphs are absent or invalid, because MARC looks for and tries to open that menugraph file whenever it gets a request for a screen that is in that menugraph. To minimize performance problems, make a compatible version of a missing or invalid menugraph available as soon as possible.

Example

The following example shows the maximum number of menugraph file titles that can be displayed by this command. Each line of this display appears only if a corresponding menugraph file has been accessed during the current MARC session. If the substitute menugraph or language menugraph is changed during the current session, only the most recently accessed menugraph appears in each case. In this example, the menugraph file (ACCOUNTING)SYSTEM/MARC/MENUGRAPH/ACCOUNTING has been removed from the family DISK.

```
Substitute: (ACCOUNTING)SYSTEM/MARC/MENUGRAPH/GENERALLEDGER ON SOMEPACK
Usercode  : (ACCOUNTING)SYSTEM/MARC/MENUGRAPH/ACCOUNTING ON DISK
           WARNING! MARC cannot find this menugraph file.
Language  : *SYSTEM/MARC/MENUGRAPH/FRANCAIS ON DISK
System    : *SYSTEM/MARC/MENUGRAPH/ESPANOL ON DISK
English   : *SYSTEM/MARC/MENUGRAPH/ENGLISH ON DISK
```

Declaring a Substitute Menugraph

The substitute menugraph, if there is one, is the first menugraph MARC searches for any requested screens. MARC enables you to declare a substitute menugraph while your session is in progress and to switch to a different substitute menugraph at any time. Use the following form of the GO action to make these changes:

```
GO <screen name> IN <menugraph file title>
```

At the start of your session, no substitute menugraph is declared. MARC searches for any requested screens in the other menugraphs described earlier in this section.

Entering the action *GO <screen name> IN <menugraph file title>* causes MARC to adopt the specified menugraph as the substitute menugraph and to display the specified screen in that menugraph. The menugraph is declared as the substitute menugraph only if the screen is found in it.

Once a substitute menugraph has been declared, the same form of the GO action can be used to declare a different menugraph as the substitute menugraph. Since there can be only one substitute menugraph at a time, declaring a different menugraph causes the former substitute menugraph to lose its substitute status.

Bypassing the Menugraph Search Order

The following actions bypass the search order described for the active menugraphs earlier in this section.

Action	Effect
GO <screen name> IN *	This action causes MARC to bypass the substitute menugraph in its search for a screen with the specified screen name. MARC searches the other menugraphs in the active menugraph list in the normal order, displaying the first occurrence that it finds.
HOme	This action always displays home menu on your usercode default menugraph. This feature enables you always to return to a familiar starting point if you are unsure of which menugraph you are in.
PRev and REturn	These actions redisplay screens from the screen cache. Thus, if the substitute menugraph was changed recently or a menugraph was recently updated by the NEW MENU command, it is possible to display a screen that cannot be found in the new menugraph by any screen traversal action (GO, PARENT, or HOME).

Note: *The same screen name can occur in different menugraphs and can refer to screens that are not related to each other. You can always specify which menugraph you want MARC to search first by using the GO <screen name> IN <menugraph file title> action to specify the substitute menugraph.*

Customizing the Help Text

Unisys supplies two help book files with the MARC product. These files are the help books for the short and long help text for the standard MARC screens. The books contain help text for every menu, menu selection, form, and form field available in MARC, together with online tutorial information. The help book files are named

BOOK/MARC/ENGLISH
BOOK/MARC/SHORTHELP/ENGLISH

To be accessible by MARC, the BOOK files must have the same usercode and must reside on the same disk family as the support library identified by naming the HELPSUPPORT library in a SL (Support Library) system command. See the *A Series System Commands Operations Reference Manual* for information on the SL command.

If you add custom menus, menu selections, fields, or forms to the MARC screens, you might also want to add help text to support them. To enable you to do so, Unisys provides the corresponding help book source files. You can customize any of the help text to suit your particular installation by modifying these files and producing customized help book files. The help book source files are named

SYMBOL/MARC/ENGLISH
SYMBOL/MARC/SHORTHELP/ENGLISH

For information on translating the help files into another language, see “Localizing the MARC Online Documentation” and “Localizing Help Text Keywords” in Section 9, “Using MARC Internationally.”

Writing Your Own Help Text

Any changes you need to make must be made to the help book source files—you cannot change the help book files directly.

To change a help book source file, use the A Series Editor or edit the file in CANDE. If you are adding new help keywords, use a FIND command to make sure that your new keyword is unique in the file. When you have completed the changes to the help book source file, run the A Series Help Utility (available from the TOOLS menu) to create the new help book, using the updated help book source file as input to the utility.

See the *A Series Help Utility Operations Guide* for more information on the layout of help book source files and the commands you can use in these files.

Making Custom Help Text Available to Users

When you have created a new help book, test the accessibility and appearance of your new help text before making the book available throughout the system. To do so, enter the following command in the *Action* field:

```
HELP <keyword> IN <new book file title> ON <family name>
```

MARC then displays your new help text. Repeat this command for all the keywords that you have added or for which you have changed the text. For more information on the HELP command, see “Using the HELP Command” in Section 3, “Using the Online Help Facilities.”

If you make changes to the same help book more than once during the same session, use the command *NEW BOOK <new book file title> ON <family name>* to ensure that the system accesses the most recently updated version. The NEW BOOK form of the NEW command is described in Section 6, “Using Commands.”

When you are satisfied with the appearance of the new text, you can make the new help book available to other users on your system. If you are replacing the book supplied by Unisys, you can delete the old book and copy the new book to the same family and usercode as the original, provided that no users are currently accessing the old book.

To determine whether anyone is accessing the help text, enter the following command in the *Action* field:

```
FILE <help book title>
```

If the book is in use, the resulting display includes the words *IN USE* on the first line.

You can use IMG to associate a different default help book with each menugraph in the active menugraph list. You can also use IMG to instruct MARC to retrieve help

text from a particular help book for any menu, menu selection, form, or form field in a menugraph.

Using COMS Processing Items with MARC

A COMS *processing item* is a procedure you can create to process a transaction message either before an application program receives the message or after the application program sends it. Processing items are useful for reformatting or filtering messages, logging messages, gathering statistics, storing information about the state of a transaction, or emulating various types of terminals.

To simplify the use of processing items with MARC, COMS provides two entities known as agendas. A COMS *agenda* is a mechanism for routing and processing messages. A processing item becomes part of an agenda when it is associated with an agenda in the COMS configuration file.

MARCINPUT and MARCOUTPUT Agendas

The agendas that COMS provides for MARC are

- An input agenda called MARCINPUT
- An output agenda called MARCOUTPUT

MARC obtains the output agenda designator for MARCOUTPUT from COMS during initialization and attaches this designator to all output, with one exception. This exception is the *WINDOW <name>/<dialogue>* message that is sent – typically to the status line – as the response to a successful ON command request. This message is sent directly through COMS rather than through the agenda mechanism because *WINDOW <name>/<dialogue>* must arrive before messages that might be queued for the destination window.

Formatting Display Output

MARC provides the library SYSTEM/MARC/AGENDA/TDXXX to format output for terminals emulating a TD830 terminal. MARC links to this library directly, and filters all output for TD-like terminals through this library before making that output available to processing items belonging to the MARCOUTPUT agenda.

On input, the processing items associated with the MARCINPUT agenda can modify only input for TD-like terminals before those items are filtered through this library. The SYSTEM/MARC/AGENDA/TDXXX library is therefore unconditionally called before any input from (or output to) COMS can be processed.

Guidelines for Writing Processing Items

A newly created processing item for MARC must be error-free. Fatal errors can cause MARC to behave unpredictably and can abnormally terminate other MARC tasks.

Creating Custom Versions of MARC

When writing processing items for MARC, be aware of the use of the *conversation area* of message headers. Table 8-1 lists the functions of the words in the conversation area. See the *A Series Communications Management System (COMS) Programming Guide* for additional information on message headers and writing processing items.

Table 8-1. Conversation Area of Message Headers

Word	Function
HDR[0] (first word)	MARC uses word 0 (the first word) of the conversation areas of both the input and output headers to contain information for each processing item concerned with screen formatting. For MARC, this word is used exclusively by the SYSTEM/MARC/AGENDA/TDXXX library.
HDR[1] (second word)	<p>If the value of word 1 (the second word) is greater than or equal to zero, but less than or equal to 65535, the value is the MultiLingual System (MLS) message number of the last MLS message MARC formatted into the output.</p> <p>If the value is less than 0, but equal to or greater than -65535, the value is a valid MLS message number of the last MLS message MARC formatted. However, either of the following situations might exist:</p> <ul style="list-style-type: none">• The MLS message is missing from the message header.• The MLS message is in a language other than the original language requested. <p>This information provides a means of using an output processing item to monitor proper translations of messages.</p> <p>A value outside the range -65535 to +65535 indicates that MARC was unable to format an MLS message number into the output. MARC places the value "400000010000" (representing -65536) into this word to represent an invalid value.</p>
HDR[2-n] (third and following words)	These words are reserved for future use.

Using Directive Commands

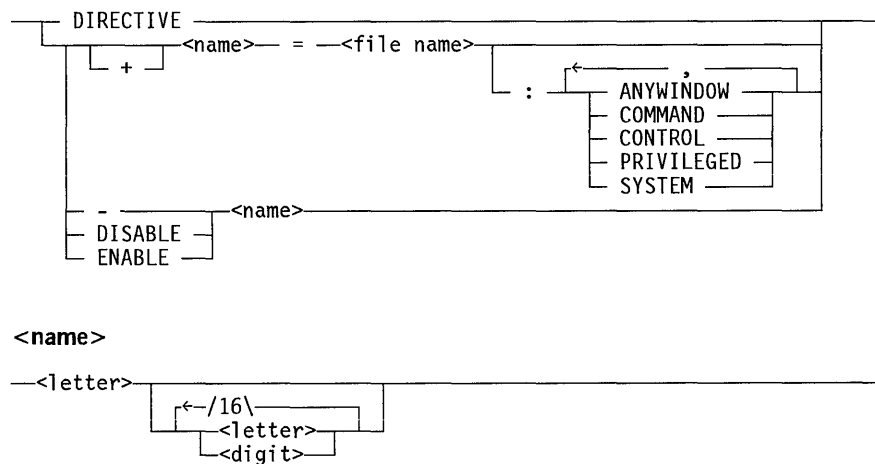
MARC provides a feature that enables you to create new commands and make them available to MARC users. Commands that you create in this way are called *directive commands*. You define the function of these commands by writing a library of ALGOL procedures. You then use the **DIRECTIVE** command in MARC to associate a command name with the procedure. Thereafter, users can use the new command in the same way as they use any other MARC command.

DIRECTIVE Command

The DIRECTIVE command associates a name with a procedure in a user-written library. When that name is used later as a command, MARC executes the associated procedure. MARC can thus execute a user-defined, user-created, and user-named process – that is, a directive – in addition to the processes provided with MARC.

The DIRECTIVE command can eliminate the need to write and define MCSs on a system. Because MARC executes the library procedure, the library procedure gains MCS privileges and capabilities. With this command, you can direct MARC to execute a number of functions that can otherwise be executed only by a specialized MCS, such as MCSLOGGER.

Syntax



Explanation

Up to 12 different directives can be available at any one time. Because a name can be followed by text, each name can result in a number of actual commands. For this reason, only users with SYSTEMUSER privileges can use the DIRECTIVE command. In addition, if the value of the system option SECADMIN is TRUE, only a usercode with security-administrator status can use the command definition features of the DIRECTIVE command. However, any user can use the DIRECTIVE command with no parameters to display the names of available directive commands.

When a command name is established, conflicts with other commands are considered to be intentional overrides. The only exceptions to this rule are the WRU command (which MARC handles in a special way) and the DIRECTIVE command itself. MARC explicitly prevents users from assigning either of these two commands as a name.

If the procedure executed by the directive halts, MARC recovers and continues execution. However, if the procedure enters an infinite loop, the MARC task that called the procedure *hangs* – that is, it stops, waiting for a response. If the processing of the procedure takes an excessive amount of time, MARC performance is degraded.

Creating Custom Versions of MARC

A colon (:) must precede a list of one or more of the options ANYWINDOW, COMMAND, CONTROL, PRIVILEGED, and SYSTEM.

DIRECTIVE

Lists all current directive commands.

DIRECTIVE <name> = <file name>
DIRECTIVE + <name> = <file name>

Assigns the specified directive command name to the command contained in the specified file. The plus (+) option is a safety measure that prevents your overwriting an existing directive. If you use this option and the name you specify already exists, MARC displays the following message:

```
<name> HAS ALREADY BEEN DEFINED
```

If you omit this option, a new name is created if none currently exists; the newly defined directive overwrites an existing directive.

DIRECTIVE - <name>

Deletes a directive command name. For example, suppose you have defined the name INFO by entering the following command:

```
DIRECTIVE INFO =SYSTEM/INFO/LIBRARY ON LIBS :SYSTEM
```

Now, however, you no longer want directives with the name INFO to be recognized. Entering *DIRECTIVE - INFO* ensures that MARC rejects commands preceded by the word INFO.

DIRECTIVE DISABLE <name>

Disables a directive command name without deleting the name.

If the name is currently enabled, specifying that name in a *DIRECTIVE DISABLE* command disables the name, delinks the process from the name if it is currently linked, and removes the process from the mix if it is currently running and is unfrozen. However, the association between the name and the process is unaffected. This command differs from the delete form (-), which eliminates the directive name entirely so that it is no longer recognized by MARC.

If the process associated with the name is currently in use – that is, if someone has invoked the name and the associated process is still active – specifying that name in a *DIRECTIVE DISABLE* command schedules the name to be disabled and delinked. It also prevents use of the name by any new users. When the process that is in progress completes, the name is disabled and delinked.

If the name is currently scheduled to be disabled, specifying that name in a *DIRECTIVE DISABLE* command has no effect on the status of the name. The name remains scheduled to be disabled and delinked. However, specifying that name in a

DIRECTIVE ENABLE command at this point causes the name to become available again.

If a name that matches a name currently scheduled to be disabled is specified in a REPLACE (Replace Disk or Pack Volume) system command, the new directive is established in the ENABLED state and is available for immediate use.

If the name is currently disabled, specifying that name in a DIRECTIVE DISABLE command causes a message to appear stating that the name is already disabled.

DIRECTIVE ENABLE <name>

Reenables a directive name that was disabled by a DIRECTIVE DISABLE command. The next invocation of the name causes the associated process to be executed and relinked. If the name is currently enabled, specifying that name in a DIRECTIVE ENABLE command causes a message to appear stating that the name is already enabled. If the name is currently scheduled to be disabled and delinked, specifying that name in a DIRECTIVE ENABLE command causes the name to be enabled and made available for use.

ANYWINDOW

Enables you to use the directive as a control-type command – that is, a command preceded by a question mark (?) – that can be entered from any window except message control system (MCS) windows such as CANDE. The directive is processed by MARC dialogue 1 (MARC/1).

***Note:** Take care when specifying an ANYWINDOW directive, because overall MARC performance can be degraded if the called procedure is inefficient or fails to return control to MARC quickly.*

COMMAND CONTROL PRIVILEGED SYSTEM

Specify security attributes for each command. These features enable you to change the security of an existing command as well as specify the security for new directives. For example, the command PS FORCE has a SYSTEMUSER security level. You can write a procedure that executes the PS FORCE command and then define a directive without the SYSTEM security attribute that executes the procedure.

The security attributes have different names in different code files. The naming conventions used by the DIRECTIVE command, the USERDATAFILE usercode database, and COMS are as follows:

DIRECTIVE Command Option	USERDATAFILE Equivalent	COMS Equivalent
COMMAND	Unavailable	COMMANDCAPABLE
CONTROL	Unavailable	CONTROLCAPABLE
PRIVILEGED	PU	PRIVILEGEDUSER

continued

Creating Custom Versions of MARC

continued

DIRECTIVE Command Option	USERDATAFILE Equivalent	COMS Equivalent
SYSTEM	SYSTEMUSER	SYSTEMUSER

Security attributes are additive. For example, if both the PRIVILEGED and SYSTEM attributes are assigned to a command, the user must be both a privileged user and a SYSTEMUSER to execute the command. If the user lacks either status, the system rejects the command and issues a security error message.

Note: *When a directive command changes the security status of the user, the newly acquired privilege applies only to the user executing the directive command, and it applies only during the execution of that directive command. Further, the new privilege is effective only in MARC and certain other components of the system software, primarily COMS and the CONTROLLER. Other components—such as the GETSTATUS intrinsic, used to locate files—use the information in the USERDATAFILE to determine privileges.*

Directives Definition File

MARC stores established directives in a directives definition file that has the following file name:

```
SYSTEM/<MCS name>/COMMANDER/DIRECTIVES
```

Storing the directives in a separate file enables sites to transfer the directive-handler libraries and the directives definition file from one system to another.

Note: *Take care when copying a new directives definition file onto a system that currently has MARC running: if the file is altered by a process other than a MARC procedure, the system rejects changes until MARC is reinitialized. Only one directives definition file is permitted on each system.*

MARC links to the libraries of directives that are defined in a directives definition file the first time a directive is invoked rather than requiring a directives definition file during initialization or automatically attempting linkage if such a file is present. This procedure speeds the MARC initialization process.

Writing Directive Commands

The following discussion provides information to help you write your own procedures that can be executed as directives.

Format of the Source Code

If more than one copy of MARC is active at a time, a directive might be required to process two or more requests at the same time. If the directive code might have difficulty operating in this multiuser environment, limit the number of MARC worker stacks. On the P screen of the COMS Utility, set to 1 the number of maximum copies of MARC.

When you execute the DIRECTIVE command, MARC links by title to an external library that contains an entry point named – or exported as – EXTERNALCOMMAND. Entry points must be defined in an ALGOL procedure that adheres to the following format:

```

INTEGER PROCEDURE EXTERNALCOMMAND(
    COMS_HEADER                % COMS HEADER ARRAY
    ,COMMAND                   % TEXT AFTER DIRECTIVE
    ,TAG                        % SECURITY, LANGUAGE
    ,DIRECTIVE                 % COMMAND PREFIX
    ,GET_DIALOG_ATTRIBUTE      % DIALOG ATTRIBUTE PROC
    ,RESPONSE_FORMATTER       % RETURN RESPONSE PROC
    ,SCRATCH                   % SCRATCH EBCDIC ARRAY
);
REAL ARRAY COMS_HEADER[*];
REAL ARRAY COMMAND[*];
REAL ARRAY TAG[*];
REAL ARRAY DIRECTIVE[*];
INTEGER PROCEDURE GET_DIALOG_ATTRIBUTE(ATTNO,ATT);
    VALUE ATTNO;
    INTEGER ATTNO;
    REAL ARRAY ATT[*];
    FORMAL;
INTEGER PROCEDURE RESPONSE_FORMATTER(TEXT_LENGTH,TEXT);
    VALUE TEXT_LENGTH;
    INTEGER TEXT_LENGTH;
    EBCDIC ARRAY TEXT[*];
    FORMAL;
EBCDIC ARRAY SCRATCH[*];

```

Parameters

The ALGOL library procedure EXTERNALCOMMAND uses the following parameters:

- COMS_HEADER
- COMMAND
- TAG
- DIRECTIVE
- GET_DIALOG_ATTRIBUTE
- RESPONSE_FORMATTER

Creating Custom Versions of MARC

- SCRATCH

These parameters are explained in the following paragraphs. See “Procedure Result” later in this section for an explanation of the values returned by the EXTERNALCOMMAND library procedure.

COMS_HEADER Parameter

This parameter is the COMS header array given to MARC by COMS. User processing items can change or add to information in this array, making access to this information desirable – or even necessary – for some sites. The format of this array is described in the *A Series Communications Management System (COMS) Programming Guide*.

COMMAND Parameter

This parameter is an array that contains any text specified after the directive, with leading and trailing blanks deleted. The first word of this array contains the length of the EBCDIC text that begins in the second word. This text is deliberately kept distinct from the directive itself. In addition, this parameter can be used to return a user-created command. See the explanation for value 4 under “Procedure Result” later in this section.

For example, suppose a user enters *MYCOMMAND ABCDE* either through a MARC menugraph selection or directly in the *Action* field. COMMAND contains the value 5 in the first word and the character string *ABCDE*, in EBCDIC, beginning in the second word (the seventh character).

TAG Parameter

This parameter is an array that contains the following specific information about the calling dialogue:

- TAG[0]

Contains security information about the calling dialogues – specifically, information representing the security state of the calling dialogue at the instant of the call – as follows:

[03:01]	1 means COMMANDCAPABLE capability.
[02:01]	1 means SYSTEMUSER capability.
[01:01]	1 means PRIVILEGED capability.
[00:01]	1 means COMS CONTROL capability.

- TAG[1]

Contains the length of the language attribute value (string) for this dialogue.

- TAG[2]

Contains the value of the language attribute (string). The length of the string is found in TAG[1].

DIRECTIVE Parameter

This parameter is an array that contains the names of the directives entered by the user. This parameter is deliberately separated from the command text. The first word of the array contains the length of the name. The name, in EBCDIC, begins in the second word (the seventh character) of the array.

GET_DIALOG_ATTRIBUTE Parameter

This parameter is a procedure that can be called within the EXTERNALCOMMAND library procedure to obtain additional information about the MARC dialogue that called the EXTERNALCOMMAND procedure. A predefined MARC dialogue attribute number is passed as the first parameter (ATTNO). If the attribute is within the range of known attributes and information about the attribute of this procedure exists, this procedure returns a result value of zero, and the current value of the attribute is returned in the array parameter (ATT).

Table 8-2 lists the result values of the GET_ATTRIBUTE procedure.

Table 8-2. GET_ATTRIBUTE Procedure Result Values

Value	Meaning
0	The attribute information is valid.
1	There is no error, but the attribute information is null or undefined.
-1	The attribute number is out of range.

Table 8-3 lists the dialogue attribute numbers and the corresponding formats of the result in the array parameter.

Table 8-3. GET_ATTRIBUTE Procedure Result Formats

Value	ATTNO Result Format
0	Caller identification. MARC stores the value 1 in the first word of the array parameter. This attribute permits use of this entry point by other software that can identify itself separately.
1	The current message control system (MCS) number of COMS as defined by the Network Definition Language II (NDLII). This value is returned in the first word of the array parameter.
2	The MARC session LANGUAGE attribute for this dialogue. The length of the LANGUAGE attribute is returned in the first word of the array. The value of the LANGUAGE attribute is returned in EBCDIC, beginning in the second word.

continued

Table 8-3. GET_ATTRIBUTE Procedure Result Formats (cont.)

Value	ATTNO Result Format
3	The version, cycle, and patch numbers of the calling MARC. Words 0, 1, and 2 contain the version, cycle, and patch number (as defined by the COMPILETIME intrinsic in ALGOL), respectively.
4	The usercode for this dialogue. The length of the usercode is in the first word of the array. The usercode is in EBCDIC, beginning in the second word.
5	The accesscode for this dialogue. If an accesscode is set for the session, the length of the accesscode is placed in the first word of the array and the accesscode, in EBCDIC, begins in the second word. If no accesscode is defined, the first word of the array contains 0 and the procedure returns an ATT value of 1.
6	The chargecode for this dialogue. If a chargecode is set for this session, the length of the chargecode is placed in the first word of the array and the chargecode, in EBCDIC, begins in the second word. If no chargecode is defined, the first word of the array contains 0 and the procedure returns an ATT value of 1.
7	The logical station number (LSN) of the data comm station to which this MARC dialogue is attached. The LSN is returned in the first word of the array. If no LSN is defined, the first word contains 0 and the procedure returns an ATT value of 1.
8	The job number of the current session of this MARC dialogue. This value is returned in the first word of the array. If no session is defined, the first word contains 0 and the procedure returns an ATT value of 1.
9	The value of the JOBSUMMARY attribute for the current session of this MARC dialogue. This value is returned in the first word of the array.
10	The current value of the JOBSUMMARYTITLE attribute of this MARC dialogue. This value is returned in the second word of the array with the length of the value in the first word. If no JOBSUMMARYTITLE is set, the first word of the array contains 0 and the procedure returns an ATT value of 1.
11	The value of the PRINTDEFAULTS attribute for this MARC dialogue. This value is returned starting in the second word of the array, with the length of the value in the first word. If no PRINTDEFAULT value has been set, the first word of the array contains 0 and the procedure returns an ATT value of 1.
12	The value of the NOJOBSUMMARYIO attribute for the current session of this MARC dialogue. If the attribute is FALSE, the first word of the array contains 0. If the attribute is TRUE, the first word contains 1.

continued

Table 8-3. GET_ATTRIBUTE Procedure Result Formats (cont.)

Value	ATTNO Result Format
13	The current value of the DESTNAME attribute for this MARC dialogue. This value is returned in EBCDIC, starting in the second word of the array, with the length of the value in the first word. If no DESTNAME is set, the first word of the array contains 0 and the procedure returns an ATT value of 1.
14	The station name of this MARC dialogue. This value is returned in EBCDIC, starting in the second word of the array, with the length of the station name in the first word.
15	The CONVENTION attribute for this dialogue. The length of this attribute is returned in the first word of the array. The value of the attribute is returned in EBCDIC, beginning in the second word.

RESPONSE_FORMATTER Parameter

This parameter is a formal procedure that is called by the EXTERNALCOMMAND library procedure once for each line of the response to be returned to MARC for communication to the user. Characters such as carriage return and line feed characters can be embedded within text to produce multiple lines of output in one call to the RESPONSE_FORMATTER procedure.

This procedure has two parameters of its own: TEXT_LENGTH and TEXT. The TEXT_LENGTH parameter represents the number of characters of text, and the text itself is contained in the TEXT parameter—the second parameter. The TEXT_LENGTH parameter must be no more than 255 characters; if it is, only the first 255 characters of the text are used.

MARC formats lines exceeding the screen width to avoid truncation of the output text. The procedure returns a value of 1 if MARC determines that formatting is necessary for one of the following reasons:

- The message is longer than the screen width.
- The message contains characters with values less than an EBCDIC *space* (hexadecimal 40) that would result in multiple lines of output.

In all other cases the result is 0.

SCRATCH Parameter

This parameter is a scratch array for use by the EXTERNALCOMMAND library procedure. This parameter minimizes the need for the library to perform buffer management. You can use this array to format the output of the call to the RESPONSE_FORMATTER procedure. For performance reasons, this array should be

Creating Custom Versions of MARC

initialized by the library when needed rather than resized or deallocated. MARC does not clear this array before calling the library entry point.

Procedure Result

Field [07:08] of the value returned by the EXTERNALCOMMAND library procedure contains a value. Each possible value returns a different message. Table 8-4 lists the values, the associated messages, and their meanings.

Field [47:16] of the value returned by the EXTERNALCOMMAND procedure is reserved for error enumerations. MARC ignores this field.

Table 8-4. EXTERNALCOMMAND Library Procedure Result Values

Value	Message and Meaning
0	<p>NO ERROR</p> <p>You used the RESPONSE_FORMATTER procedure, described earlier in this section, to format messages. If no messages have been formatted, MARC produces messages indicating that no output messages were generated by the input.</p>
1	<p>GENERAL SYNTACTICAL ERROR</p> <p>The command contains an error. This message is MARC MLS message number 3004.</p>
2	<p>RESERVED</p> <p>This value is currently unused.</p>
3	<p>REESTABLISH SECURITY AND CONTINUE</p> <p>MARC rechecks the SECURITY DESIGNATOR word in the COMS_HEADER array parameter for changes to this user's security capabilities. MARC then uses the original user input. The SECURITY DESIGNATOR word overrides settings in the USERDATAFILE usercode database settings instead of complementing them, as is normal in MARC. The SECURITY DESIGNATOR word already contains values which reflect the security of the station as defined in the COMS configuration file, even if the directive does not change the word.</p> <p>This feature enables you to monitor particular commands to further restrict or to remove the restrictions on the use of the command for particular users. If the user has formatted messages using RESPONSE_FORMATTER, these messages appear before any messages generated as a result of the eventual command processing.</p>
4	<p>REESTABLISH SECURITY AND USE NEW COMMAND</p> <p>MARC rechecks the SECURITY DESIGNATOR word in the COMS_HEADER array parameter as in value 3, but command interpretation continues with a command formatted by the user in the COMMAND array parameter.</p> <p>Also as in value 3, the SECURITY DESIGNATOR word overrides settings in the USERDATAFILE instead of complementing them, as is normal with MARC.</p>

continued

Table 8-4. EXTERNALCOMMAND Library Procedure Result Values (cont.)

Value	Message and Meaning
5	<p>The length of the generated command must appear in word 0 (the first word) and the command itself must appear in EBCDIC starting in word 1. The length actually accepted by MARC is the lower of 1200 or the length in word 0. The length of the generated command should not, therefore, exceed 1200 characters. Messages formatted by the RESPONSE_FORMATTER appear before any messages generated as a result of the eventual command processing.</p> <p>LEAVE SECURITY AS IS AND USE NEW COMMAND</p> <p>MARC does not check the SECURITY DESIGNATOR word in the COMS_HEADER array parameter, but command interpretation continues with a command formatted by the user in the COMMAND array parameter, as in value 4.</p> <p>The SECURITY DESIGNATOR word does not override the user's existing USERDATAFILE security.</p>
6	<p>The length of the generated command must appear in word 0 (the first word) and the command itself must appear in EBCDIC starting in word 1. The length actually accepted by MARC is the lower of 1200 or the length in word 0. The length of the generated command should not, therefore, exceed 1200 characters. Messages formatted by the RESPONSE_FORMATTER appear before any messages generated as a result of the eventual command processing.</p> <p>COMMAND ERROR</p> <p>You have formatted error messages using the RESPONSE_FORMATTER procedure. If no messages were formatted, MARC generates messages indicating that no messages were generated by the input.</p>

Sample Directive Command

The following is an example of a source library procedure that uses the **DIRECTIVE** command. An explanation of the procedure and its results follows the code.

Creating Custom Versions of MARC

```
BEGIN
INTEGER PROCEDURE EXTERNALCOMMAND (
    COMS_HEADER,           % COMS Header
    COMMAND,               % Command Text after Directive
    TAG,                   % Security, Language
    DIRECTIVE,             % Directive
    GET_DIALOG_ATTRIBUTE,  % Get Attribute
    RESPONSE_FORMATTER,    % Return Response Procedure
    SCRATCH);              % Scratch Array
REAL ARRAY COMS_HEADER[*];
REAL ARRAY COMMAND[*];
REAL ARRAY TAG[*];
REAL ARRAY DIRECTIVE[*];
INTEGER PROCEDURE GET_DIALOG_ATTRIBUTE (ATTNO, ATT);
    VALUE ATTNO;
    INTEGER ATTNO;
    REAL ARRAY ATT[*];
    FORMAL;
INTEGER PROCEDURE RESPONSE_FORMATTER (TEXT_LENGTH, TEXT);
    VALUE TEXT_LENGTH;
    INTEGER TEXT_LENGTH;
    EBCDIC ARRAY TEXT[*];
    FORMAL;
EBCDIC ARRAY SCRATCH[*];
```

```

BEGIN

POINTER PSCR;
INTEGER RSLT;

REAL ARRAY RSCRATCH [0:59];
DEFINE      % Attribute Numbers
  ATT_CALLERV      = 0 #, % Type of caller (1=MARC)
  ATT_MARC_TAGV    = 1 #,
  ATT_MCSNRV      = 1 #, % MCS number of calling procedure
  ATT_LANGUAGEV   = 2 #, % Length of language string
  ATT_VERSIONV    = 3 #, % MARC version cycle and patch
  ATT_USERCODEV   = 4 #, % Usercode
  ATT_ACCESSCODEV = 5 #, % Accesscode
  ATT_CHARGEDEV   = 6 #, % Chargecode
  ATT_LSNV        = 7 #, % LSN
  ATT_SESSIONV    = 8 #, % Session job number
  ATT_JOBSUMMARYV = 9 #, % Jobsummary
  ATT_JOBSUMMARYTITLEV = 10#, % Jobsummary title
  ATT_PRINTDEFAULTSV = 11#, % Print defaults
  ATT_NOJOBSUMMARYIOV = 12#, % Nojobsummary I/O
  ATT_DESTNAMEV   = 13#, % Destination name
  ATT_STATION_NAMEV = 14#, % Source station name
  ATT_CONVENTION  = 15#, % Convention
  ATT_MAX_MAX     = 15#; % Maximum attribute number

DEFINE      % Function results
  ATTRES_NULL_VALUEV = 1#, % Null results (not for dialogue)
  ATTERR_INV_FUNCTIONV = -1#; % Function out of range

DEFINE      % Procedure results
  RSLT_NO_ERROR      = 0#, % No error
  RSLT_SYNTAX_ERROR  = 1#, % General syntax error
  RSLT_RESERVED      = 2#, % Reserved for Unisys
  RSLT_NEWSEC        = 3#, % Reestablish new security and
                          % continue with directive command
  RSLT_NEWSEC_COMMAND = 4#, % Reestablish new security and
                          % continue with command
  RSLT_NEWCOMMAND    = 5#; % Leave security and use command
DEFINE      % Miscellaneous
  COMMA              = ,#,
  SAY(WHATEVER)     =
    BEGIN
      REPLACE PSCR:SCRATCH[0] BY WHATEVER;
      RESPONSE_FORMATTER (OFFSET (PSCR), SCRATCH);
    END#,

  DIR_COMMANDCAPABLE = 3 #, % [03:01] security word
  DIR_SYSTEMUSER     = 2 #, % [02:01] security word
  DIR_PRIVILEGEDUSER = 1 #, % [01:01] security word
  DIR_CONTROLCAPABLE = 0 #; % [00:01] security word

```


Creating Custom Versions of MARC

```
LABEL XIT;

%%% Create a directive 'GO' to disable the MARC 'GO' action      %%%
%%% for all users. Using SYSTEM/IMG to disable this action      %%%
%%% disables 'GO' only for nonprivileged users, which could     %%%
%%% cause problems for menugraphs with built-in 'GO' commands.  %%%

IF POINTER(DIRECTIVE[1]) = "GO" THEN
  BEGIN
    SAY ("Sorry, 'GO' is not a valid command on this system.");
    GO XIT;
  END;

%%% Directives 'INFO' and 'MORE' use RESPONSE_FORMATTER to      %%%
%%% report security/dialogue data for this session/station.     %%%

IF POINTER(DIRECTIVE[1]) = "INFO" THEN
  BEGIN
    SAY ("This information is for your current MARC dialogue: "
        COMMA 48"0D25" COMMA " ");
    RSLT:=GET_DIALOG_ATTRIBUTE (ATT_STATION_NAMEV, RSCRATCH);
    IF (RSLT GEQ 0) AND (RSCRATCH[0] GTR 0) THEN
      SAY (" Station name          : " COMMA POINTER(RSCRATCH[1])
          FOR RSCRATCH[0]);
    RSLT:=GET_DIALOG_ATTRIBUTE (ATT_USERCODEV, RSCRATCH);
    IF (RSLT GEQ 0) AND (RSCRATCH[0] GTR 0) THEN
      SAY (" Usercode of caller   : " COMMA POINTER(RSCRATCH[1])
          FOR RSCRATCH[0]);
    RSLT:=GET_DIALOG_ATTRIBUTE(ATT_ACCESSCODEV, RSCRATCH);
    IF (RSLT GEQ 0) AND (RSCRATCH[0] GTR 0) THEN
      SAY (" Accesscode of caller : " COMMA POINTER(RSCRATCH[1])
          FOR RSCRATCH[0]);
    RSLT:=GET_DIALOG_ATTRIBUTE(ATT_CHARGECODEV, RSCRATCH);
    IF (RSLT GEQ 0) AND (RSCRATCH[0] GTR 0) THEN
      SAY (" Chargecode of caller : " COMMA POINTER(RSCRATCH[1])
          FOR RSCRATCH[0]);
    IF TAG[0] IS 0 THEN
      SAY (" User has no SYSTEM, PRIVILEGED, or CONTROL status.")
    ELSE
      SAY (" The following security capabilities are available:");
    IF BOOLEAN(TAG[0]).[DIR_SYSTEMUSER:01] THEN
      SAY (" SYSTEMUSER      Capabilities");
    IF BOOLEAN(TAG[0]).[DIR_PRIVILEGEDUSER:01] THEN
      SAY (" PRIVILEGED      Capabilities");
    IF BOOLEAN(TAG[0]).[DIR_CONTROLCAPABLE:01] THEN
      SAY (" CONTROLCAPABLE Capabilities");
    IF BOOLEAN(TAG[0]).[DIR_COMMANDCAPABLE:01] THEN
      SAY (" COMMANDCAPABLE Capabilities");
    SAY (" ");
    SAY ("Enter 'MORE' action to continue dialogue information.");
    GO XIT;                % No result returned. Use default.
  END;
```

```

IF POINTER(DIRECTIVE[1]) = "MORE" THEN
  BEGIN
  SAY ("The following information is current for this dialogue:"
      COMMA 48"ØD25" COMMA " ");
  RSLT:=GET_DIALOG_ATTRIBUTE (ATT_CALLERV, RSCRATCH);
  IF RSLT GEQ Ø THEN
    IF RSCRATCH[Ø] EQL 1 THEN
      SAY (" Calling program      : MARC")
    ELSE
      SAY (" Calling program      : " COMMA RSCRATCH[Ø]
          FOR * DIGITS COMMA " (UNKNOWN CALLER ID)");
  SAY (" Directive passed       : " COMMA POINTER(DIRECTIVE[1])
      FOR DIRECTIVE[Ø]);
  IF COMMAND[Ø] GTR Ø THEN
    SAY (" Command information   : " COMMA POINTER(COMMAND[1])
        FOR COMMAND[Ø]);
  SAY (" User language         : " COMMA POINTER(TAG[2])
      FOR TAG[1]);
  RSLT:=GET_DIALOG_ATTRIBUTE (ATT_STATION_NAMEV, RSCRATCH);
  IF (RSLT GEQ Ø) AND (RSCRATCH[Ø] GTR Ø) THEN
    SAY (" Station name         : " COMMA POINTER(RSCRATCH[1])
        FOR RSCRATCH[Ø]);
  RSLT:=GET_DIALOG_ATTRIBUTE(ATT_VERSIONV, RSCRATCH);
  IF RSLT GEQ Ø THEN
    SAY (" MARC version         : " COMMA RSCRATCH[Ø] FOR * DIGITS
        COMMA "." COMMA RSCRATCH[1] FOR * DIGITS
        COMMA "." COMMA RSCRATCH[2] FOR * DIGITS);
  RSLT:=GET_DIALOG_ATTRIBUTE(ATT_MCSNRV, RSCRATCH);
  IF RSLT GEQ Ø THEN
    SAY (" COMS/MARC MCS number : " COMMA RSCRATCH[Ø]
        FOR * DIGITS);
  RSLT:=GET_DIALOG_ATTRIBUTE(ATT_LSNV, RSCRATCH);
  IF RSLT GEQ Ø THEN
    SAY (" LSN of station       : " COMMA RSCRATCH[Ø]
        FOR * DIGITS);
  RSLT:=GET_DIALOG_ATTRIBUTE(ATT_SESSIONV, RSCRATCH);
  IF RSLT GEQ Ø THEN
    SAY (" Jobnumber of session : " COMMA RSCRATCH[Ø]
        FOR * DIGITS);
  RSLT:=GET_DIALOG_ATTRIBUTE(ATT_JOBSUMMARYV, RSCRATCH);
  IF RSLT GEQ Ø THEN
    SAY (" Jobsummary attribute : " COMMA RSCRATCH[Ø]
        FOR * DIGITS);
  RSLT:=GET_DIALOG_ATTRIBUTE(ATT_JOBSUMMARYTITLEV, RSCRATCH);
  IF (RSLT GEQ Ø) AND (RSCRATCH[Ø] GTR Ø) THEN
    SAY (" Jobsummary title     : " COMMA POINTER(RSCRATCH[1])
        FOR RSCRATCH[Ø]);
  RSLT:=GET_DIALOG_ATTRIBUTE(ATT_PRINTDEFAULTSV, RSCRATCH);
  IF (RSLT GEQ Ø) AND (RSCRATCH[Ø] GTR Ø) THEN
    SAY (" Printdefaults        : " COMMA POINTER(RSCRATCH[1])
        FOR RSCRATCH[Ø]);
  )

```

Creating Custom Versions of MARC

```

RSLT:=GET_DIALOG_ATTRIBUTE(ATT_DESTNAMEV, RSCRATCH);
IF (RSLT GEQ 0) AND (RSCRATCH[0] GTR 0) THEN
  SAY (" Destination name      : " COMMA POINTER(RSCRATCH[1])
      FOR RSCRATCH[0]);
GO XIT;          % No result returned.  Use default.
END;
%%% Directive 'QUIT' disables the 'QUIT COMS' command from      %%%
%%% MARC for COMS-control-capable users.                        %%%

IF POINTER(DIRECTIVE[1]) = "QUIT" THEN % No one can do QUIT COMS!
  BEGIN
    %%% Turn off CONTROL for this command.  MARC processes the %%%
    %%% directive as a command after reestablishing security. %%%
    COMS_HEADER[3].[00:01] := 0;
    EXTERNALCOMMAND := RSLT_NEWSEC;
    GO XIT;
  END;

%%% This directive enables non-SYSTEMUSER users to enter      %%%
%%% inquiries in MARC by prefixing system commands with 'ODT'. %%%

IF POINTER(DIRECTIVE[1]) = "ODT" THEN
  IF POINTER(COMMAND[1]) = "A"      OR
    POINTER(COMMAND[1]) = "C"      OR
    POINTER(COMMAND[1]) = "S"      OR
    POINTER(COMMAND[1]) = "W"      OR
    POINTER(COMMAND[1]) = "U"      OR
    POINTER(COMMAND[1]) = "MCS"    OR
    POINTER(COMMAND[1]) = "DBS"    OR
    POINTER(COMMAND[1]) = "LIBS"    OR
    POINTER(COMMAND[1]) = "MSG"    OR
    POINTER(COMMAND[1]) = "PER MT" OR
    POINTER(COMMAND[1]) = "PER PK" OR
    POINTER(COMMAND[1]) = "PS SH" THEN
    BEGIN
      %%% Turn on SYSTEMUSER for command.  Let MARC do the      %%%
      %%% work with the new command.  That is, drop 'ODT' and  %%%
      %%% reestablish security.                                   %%%
      COMS_HEADER[3].[40:01] := 1;
      EXTERNALCOMMAND := RSLT_NEWSEC_COMMAND;
      GO XIT;
    END;

XIT:

END EXTERNALCOMMAND;

EXPORT EXTERNALCOMMAND;
FREEZE (TEMPORARY);

END.
```

The preceding ALGOL source code is a procedure that displays various information messages when a user executes the corresponding directive commands. One of the commands is to have the name INFO and another is to have the name MORE. The source code resides in the file named INFO/LIBRARY. Compiling that file produces the library code file OBJECT/INFO/LIBRARY.

To establish the EXTERNALCOMMAND library procedure as a directive, ask an operator with SYSTEMUSER privileges to enter the following command:

```
DIRECTIVE+INFO = OBJECT/INFO/LIBRARY
```

Then, when a user enters *INFO* in the *Action* field of a MARC screen, a screen appears containing information similar to the following:

Response returned at 09:25 AM:

The following information is current for this dialogue:

```
Station name      : ODT/1
Usercode of caller : TESTUSER
Accesscode of caller : MARK
Chargecode of caller : MARK
The following security capabilities are available:
  SYSTEMUSER      capabilities
  PRIVILEGED      capabilities
  CONTROLCAPABLE  capabilities
  COMMANDCAPABLE  capabilities
```

Enter 'MORE' action to continue dialogue information.

INFO

Creating Custom Versions of MARC

If the user then enters *MORE*, a screen resembling the following appears:

Response returned at 09:26 AM:

The following information is current for this dialogue:

```
Calling program      : MARC
Prefix passed       : MORE
User language       : ENGLISH
Station name        : ODT/1
MARC version        : 38.130.309
COMS/MARC MCS number : 2
LSN of station      : 140
Jobnumber of session : 1180
JOBSUMMARY attribute : 0
JOBSUMMARY title    : MARC/TITLE
Printdefaults      : (DESTINATION= "STA PA110 AT MICROA)
Destination name    : MARCDEST
```

MORE

The user can obtain this second screen of information – that is, the MORE screen – only by first entering *INFO* in the *Action* field.

Improving MARC Performance

You can improve the performance of MARC by increasing the number of MARC copies and by relocating two internal MARC files.

Increasing the Number of Concurrent MARC Copies

You can improve the performance of MARC by increasing the number of copies of the MARC program that run concurrently. The number of running copies is determined by COMS and always falls between minimum and maximum numbers.

The Program entity of the COMS Utility enables you to display the current values of the minimum and maximum numbers and then to modify those values. The default maximum is 2.

Note: *Increasing the maximum number of MARC program copies improves MARC performance, but only to a point. Increasing the maximum number of copies beyond a certain level actually results in a degradation in performance. Experiment with various maximums, increasing the number by 1 each time, until you determine the value that provides optimum performance.*

For further information on the COMS Utility, see the *A Series Communications Management System (COMS) Configuration Guide*.

Moving Internal Files

Another way to improve MARC performance is to move two MARC internal files to disk families other than the default family.

The internal files are

- TANKFILE, an internal file that is used for the MARC buffering system
- DIALOGINFO, an internal file that MARC uses to manage sessions

By default, these files reside on the same disk family as the file that is the main body of MARC: SYSTEM/MARC/COMMANDER. Changing the FAMILY specification for these files can reduce disk contention and enhance performance, especially during initialization. The default family is used if no family name is specified.

To change the FAMILYNAME of both the TANKFILE and the DIALOGINFO file, you can use the WFL *MODIFY* statement to modify the SYSTEM/MARC/COMMANDER file, as illustrated in the following sample WFL job:

```
BEGIN JOB MODIFY/MARC/COMMANDER;  
MODIFY *SYSTEM/MARC/COMMANDER;  
FILE TANKFILE (FAMILYNAME = <family name>);  
FILE DIALOGINFO (FAMILYNAME = <family name>);  
END JOB;
```

Neither the TANKFILE nor the DIALOGINFO file can be equated to the family name DEFAULTFAMILY.

Caution

The DIALOGINFO file should never be accessed by any user other than MARC itself. MARC does not initialize if the DIALOGINFO file is already open at the time MARC attempts to access it.

Section 9

Using MARC Internationally

A Series system interfaces can be customized so that system output can appear in a variety of languages, use multiple character sets, and follow various conventions. For example, the same A Series system can present MARC screens in English, following standard United States conventions, while simultaneously presenting another user with MARC screens in Danish, following Danish conventions.

Your installation might have received a special version of MARC, known as a localized version. The term *localization* refers to the process of adapting a product to comply with the standards and functional requirements of a given locality or line of business, and includes the process of translation.

Contact your Unisys representative for details on localized versions of MARC.

If you want to create your own localized versions of MARC, see “Creating Localized Versions of MARC” later in this section for a description of the tools provided for that purpose.

Localization can affect other A Series products and user-created products running on an A Series system in addition to MARC. The discussion of localization in this guide, however, focuses on localization as it applies to MARC. Some of the operations dealing with localization that you can perform in MARC—setting the language to be used in displaying system output throughout the system, for example—can affect other products as well as MARC. This guide describes how you perform these operations in MARC, but it does not provide detailed information on how localization applies to other products.

Consult your system administrator to determine which products on your system have been localized, and see appropriate documentation for details on how localization affects those products. For example, suppose your installation has used SDF Plus to create an application product. For information on how localization affects your application product, you would see the *A Series Screen Design Facility Plus (SDF Plus) Installation and Operations Guide* and the documentation your installation has provided for the application.

When a localized version of MARC is available on your system, you must set several related options to take advantage of the localization. The following paragraphs discuss how to set these options.

Setting Options to Support Localized Versions of MARC

If one or more localized versions of MARC have been provided for your use, you can use MARC to select a number of options relating to the use of localized versions of MARC. These options enable you to choose

Using MARC Internationally

- The coded character set and associated application rules – known as a *ccsversion* – to be used for the processing of your data
- The display language for screens and messages
- The conventions for displaying items such as dates and times

To display the current settings of these options, enter the system command SYSOPS (System Options) in the *Action* field of a MARC screen. For detailed information about this command, see the *A Series System Commands Operations Reference Manual*.

If your usercode has SYSTEMUSER status, you can change these options at both the system and the session levels.

- Changing an option at the *system level* means that the option setting applies across the entire A Series system.
- Changing an option at the *session level* means that the option setting applies only in the current COMS dialogue for the current usercode – in your current MARC/1 or MARC/2 dialogue, for example.

The *ccsversion* option can be changed only at the system level; it cannot be changed – by anyone, regardless of security status – at the session level. If your usercode is without SYSTEMUSER status, you can change the language and convention options only at the session level.

Choosing a Ccsversion

A *ccsversion* is a set of instructions for the application of a particular coded character set.

Understanding the Elements of a Ccsversion

A *ccsversion* involves two elements: a coded character set for representing data, and a set of rules to be applied for processing that character set.

- Coded character set

A *character set* contains the letters, numbers, symbols, and special characters used to represent data in a particular locality or line of business. A given character set can be useful for many environments, such as different countries or different divisions of a company. For example, one character set might contain the characters used in most Western European countries, and another might contain the characters used in Japan.

A *coded character set* is a character set in which each character is assigned a code value for ease of processing. A given character set can be encoded in several different ways to produce several different coded character sets.

- Processing rules

The processing rules defined by a *ccsversion* determine such factors as how character data is presented, collated, and cased, as well as character type: alphabetic, numeric, or blank.

You can associate a given coded character set with multiple ccsversions, but each ccsversion applies to just one coded character set.

To get the full benefit of a given ccsversion, your presentation device must be capable of displaying the character set of that ccsversion, and in the manner dictated by its processing rules. If you select the Spanish ccsversion, for example, your presentation device must be capable of presenting the necessary Spanish accents and punctuation marks.

Unisys assigns to each ccsversion both a unique name and a unique number. You can use either the name or the number to identify a ccsversion.

For additional information on ccsversions, see the *A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide*.

Setting the Ccsversion (System Level Only)

System administrators, as well as anyone permitted to use the system console, can set the ccsversion at the system level. The ccsversion cannot be set—by anyone—at any other level.

To set the system ccsversion, take any one of the following actions:

- Enter either of the following commands in the *Action* field of a MARC screen:

```
SYSOPS CCSVERSION <ccsversion name>  
SYSOPS CCSVERSION <ccsversion number>
```

For information on this command, see the *A Series System Commands Operations Reference Manual*.

- Enter *GO SYSTEMCCSVERSION* in the *Action* field of a MARC screen. Enter the new language name in the first field and transmit the screen.
- Select SYS from the MARC home menu to access the SYS system control menu, choose the LOCAL menu, and then select CCSVSN. Enter the new language name in the first field and transmit the screen.

The ccsversion name or number must be that of a ccsversion available on the system. To request a list of the available ccsversion names and numbers, enter *SHOW CCSVERSIONS* in the *Action* field of a MARC screen.

The selected ccsversion becomes the default ccsversion for the system after the next halt/load. Only one ccsversion can be active on the host system at one time.

Once set, the system ccsversion becomes the default for all users and all programs on the system. Changing the system default ccsversion to something other than its initial setting affects the operation of programs that use it. For example, a program that compares or sorts data might produce different results with different ccsversions.

Changing the ccsversion affects the set of characters MARC accepts as valid input. For example, when the standard ccsversion, *Aseriesnative*, is active, MARC accepts the

Using MARC Internationally

standard United States English alphabet. If you change the `ccsversion` to Switzerland, MARC then accepts additional Swiss characters.

A change in the `ccsversion` also causes MARC to follow new rules in processing its characters. For example, a new `ccsversion` might cause MARC to change the way it translates lowercase letters to uppercase letters.

You cannot set the `ccsversion` at the usercode, task, or session levels. However, an application program can access alternative `ccsversion`s by using facilities provided in the CENTRALSUPPORT library. The `ccsversion` accessed by an application program is active only for the duration of the program and only within the program. For information on the CENTRALSUPPORT library, see the *A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide*.

Choosing a Language

MARC enables you to choose the language in which screens, messages, and help text are displayed on your system, provided that more than one language is available on that system.

If an alternate language has been made available for use on your system, the following components might have been provided as well:

- A language name that has been bound to the MCP
- A language name that has been bound to MARC
- A menugraph containing MARC screens that have been translated to that language

The menugraph is contained in a file that has a title in the following form:

```
SYSTEM/MARC/MENUGRAPH/<language name>
```

The language name is the name of the alternate language. For example, a menugraph containing MARC screens translated into German might be stored in a file with the following name:

```
SYSTEM/MARC/MENUGRAPH/DEUTSCH
```

The language name in the menugraph file title—Deutsch, in this example—matches a language name that has been bound to MARC.

If messages have been translated into the alternate language, you might also be provided with a file containing the translated messages. For example, a file named DEUTSCHMSGs might contain MARC messages translated into German.

The following paragraphs describe how to choose a language at the system, session, usercode, and task levels. These descriptions are followed by a discussion of how MARC determines which messages to use at any given point.

Setting the System Language

The system language is the default language for the display of all screens, menus, messages, and help information throughout an entire A Series system. System administrators, as well as anyone permitted to use the system console, can set the language at the system level.

For detailed information about the system language in relation to the system as a whole—how the language is set and how it affects the system—see the *A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide*. The discussion presented here is limited to information about the system language in relation to MARC in particular.

Setting the System Language in MARC

You can set the system default language by using either MARC menus and screens or the system command SYSOPS (System Options).

Regardless of the method you use to specify the language, for consistent system results, the name you specify must exactly match a name that has been bound to the MCP software. For example, if you specify the name FRENCH but the language has been bound to the MCP as FRANCAIS, the MCP continues to translate the messages it produces into English, even though MARC (and possibly other applications) displays the FRENCH versions of the messages it produces.

To request a display of the languages that have been bound to the MCP, enter *SHOW LANGUAGES* in the *Action* field of a MARC screen. The process by which a language is bound to the MCP is outlined in the *A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide*.

To set the system language, perform one of the following steps:

- In the *Action* field of a MARC screen, enter

```
SYSOPS LANGUAGE <language name>
```

For information on the SYSOPS command, see the *A Series System Commands Operations Reference Manual*.

- Enter *GO SYSTEMLANGUAGE* in the *Action* field of a MARC screen. Enter the new language name in the first field and transmit the screen.
- Select SYS from the MARC home menu to access the SYS system control menu, choose the LOCAL menu, and then select LANG. Enter the new language name in the first field and transmit the screen.

How the System Language Affects MARC

MARC attempts to use the language you have specified as the system default language after the next halt/load. For this attempt to be successful, you must have previously translated messages into that language and then bound the translated messages to MARC.

After the next halt/load or reinitialization of MARC, the selected language becomes the MARC default language. If messages translated into the language have been bound to the MCP as well, that language also becomes the default language for the system after the next halt/load.

That is, following the next halt/load or reinitialization of MARC, MARC displays

- Messages from the set of messages that have been translated into the selected language. If no translated version of a particular message exists, MARC displays the English version of that message.
- Screens from the menugraph file for the specified language. If no corresponding menugraph file is found, MARC suspends operation until the user provides appropriate responses to RSVP prompts. If a menugraph file for the specified language exists but no translated version of a particular screen exists, MARC displays the screen from the system default menugraph file, SYSTEM/MARC/MENUGRAPH/ENGLISH.

If MARC's attempt to use the specified system default language fails on any count, the standard language – English – is used as the default language for MARC.

Setting the MARC Session Language

Any user can select a language for a specific MARC session at the session level without affecting the language of any other session. That is, selecting a language for your MARC session does not set the language for your CANDE session, setting a MARC session language for your MARC/1 session does not set the MARC session language for your MARC/2 session, and so on.

The language name must exactly match a name that has been bound to the MARC software. For example, you cannot use the name FRENCH if the language has been bound to MARC as FRANCAIS.

You can set a MARC session language by using either of two methods, as follows:

- You can enter a LANGUAGE command as described in “Commands Available in MARC” in Section 6, “Using Commands.”
- You can use MARC screens and menus by taking the following steps:
 1. Select *SC* on the MARC home menu to access SC session control menu.
 2. Enter *SSLANG* in the *Choice* field. MARC displays the SSLANG show/set language menu.
 3. Enter *SLANG* in the *Choice* field of the SSLANG menu. MARC displays the LANGUAGE set session language screen.
 4. Type the name of the language you want to use in the *Language To Be Used* field.

With either of these methods, the new session language becomes effective immediately.

For the duration of the session, the MARC session language overrides the default system language and any language selected at the usercode level. (The procedure for setting a language at the usercode level is described later in this section.) However, once the session ends, MARC uses the language specified at the usercode level. If there is no usercode-level language, MARC uses the language specified at the system level.

Note: *If English screens are acceptable, but you want MARC messages to appear in another language, bind translated messages to the MARC software, but do not translate the MARC menugraph.*

Setting the COMS Application Session Language

The *COMS application session language* is the language that COMS associates with a usercode for a specific session. It is the language returned when a language request is made of COMS. (It is also returned, along with the MARC session language, when a language request is made of MARC.)

The COMS application session language enables users to access foreign-language applications through MARC without translating MARC. MARC processes the COMS application session language only to pass requests for the language directly to COMS.

If a language is defined at the usercode level, that language becomes the COMS application session language. If the language has been bound to MARC, that language also becomes the MARC session language; otherwise, the system default language becomes the MARC session language.

To define a specified language as the COMS application session language, a user must issue a LANGUAGE command from the MARC/1 dialogue. (If the user enters the command from a MARC dialogue other than MARC/1, only the MARC session language for the corresponding MARC dialogue is set; the COMS language is unchanged, as are the session language settings for any other current MARC sessions.)

If the language named in the LANGUAGE command has been defined to MARC, that language also becomes the MARC session language; otherwise, the MARC session language remains unchanged.

Once the session ends, the COMS application session language is no longer in effect and must be reset for any new MARC session.

Setting the Language at the Usercode Level

The MAKEUSER utility enables the system administrator to associate a language other than the system default language with an individual usercode. Doing so enables a user to view MARC menus in German (for example) immediately after logging on, even if the system default language is French, as long as MARC has access to German screens and messages. For information on the MAKEUSER utility, see the *A Series Security Administration Guide*.

Setting the Language for a Task

An individual task can use the LANGUAGE task attribute to specify a language. This method overrides the language currently in effect. However, the specified language remains in effect only for the user running that task, and only until the task terminates. See the *A Series Task Attributes Programming Reference Manual* and the *A Series Task Management Programming Guide* for information about the LANGUAGE task attribute.

Most task and job messages displayed on the TASKSTATUS screen, or displayed as the result of an MSG command, appear in the language of the task that generated the message. If the task language is unavailable, the system default language is used.

How MARC Determines the Language to Use for Messages

Translated messages created and bound to MARC are not associated with particular menugraphs. As a result, displaying a screen from a menugraph in a particular language does not automatically guarantee that MARC messages are displayed in the same language. Also, some portions of the screen display – prompts and messages appearing on lines 2 and 3 – are drawn from the messages and are unaffected by the menugraph in use. Those portions of the screen might be in a different language.

The default language for MARC is English. If this default language is not overridden, any MARC messages displayed are the English language messages supplied with MARC. You can override the default language at the system, session, usercode, or task level, as discussed in the preceding paragraphs.

If a language other than English becomes the active language, MARC determines whether at least the OK message (message 0) has been translated to that language and bound to the MARC software. If so, all MARC messages that are displayed are drawn from the MARC messages for that language. Otherwise, MARC displays its default messages, which are in English.

MARC displays most system messages resulting from an ALLMSG command in the current MARC session language. If this language is unavailable in the MCP for a particular message, the message is given in the system default language.

System messages are stored as messages in the MCP, and can be translated using MSGTRANS. For information on translating system messages, see the *A Series Message Translation Utility (MSGTRANS) Operations Guide*.

MARC requests from the MCP the response to system commands in the current MARC session language. If this language is unavailable in the MCP, the response is given in the system default language.

Choosing a Convention

A *convention* is an agreed-upon format for presenting information such as dates, times, or currencies. Conventions vary according to factors such as country, line of business,

and culture. For example, in some countries it is a convention to use a period (.) for the decimal sign; in other countries a comma (,) is used.

On an A Series system, each convention has a unique name and contains definitions for date, time, numeric, monetary, and page size specifications. By selecting a convention, you select all its definitions. For more information on conventions, see the *A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide*.

Unisys supplies a number of standard conventions that are read into memory during system initialization. Standard conventions cannot be altered. However, installations can create customized conventions, which are stored in the same files and processed by the same utility as the standard conventions.

System administrators, as well as anyone permitted to use the system console, can access the CONVDF screen to view, add, modify, or delete a customized convention.

To display the CONVDF screen, select SYS from the MARC home menu to access the SYS system control menu. Choose the LOCAL menu and type *CONVDF* in the *Choice* field. From the CONVDF screen, you can perform the following operations:

- View the elements in a convention definition.
- Create a customized convention definition.
- Modify customized convention definitions.
- Delete customized convention definitions.

See the *A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide* for detailed instructions on how to perform these operations.

The following paragraphs describe how to choose a convention at the system, session, usercode, and task levels.

Setting the System Convention

System administrators, as well as anyone permitted to use the system console, can select a convention at the system level by using the SYSOPS (System Options) system command. The selected convention immediately becomes the default convention for the system.

The convention name must be the name of an available convention on the system. To request the names of the available conventions, you can take either of the following courses of action:

- Enter *SHOW CONVENTIONS* in the *Action* field of a MARC screen.
- Access the SYS system control menu from the MARC home menu, access the LOCAL menu, and then select CONV to access the SYSTEMCONVENTION change system convention screen. This screen enables you to display the available conventions on the system or to change the current system default convention.

Using MARC Internationally

To set the system convention in MARC, perform one of the following steps:

- In the *Action* field of a MARC screen, enter

SYSOPS CONVENTION <convention name>

For information on the SYSOPS (System Options) system command, see the *A Series System Commands Operations Reference Manual*.

- Enter *GO SYSTEMCONVENTION* in the *Action* field of a MARC screen to access the SYSTEMCONVENTION change system convention screen. Enter the new convention name in the first field and transmit the screen.

Setting the Session Convention

Any user can select a convention for a specific session at the session level without affecting the convention of any other session. That is, selecting a convention for your MARC session does not select the convention for your CANDE session, selecting the convention for your MARC/1 session does not select the convention for your MARC/2 session, and so on.

For example, suppose that the active convention for the session you have started on MARC/1 is *Aseriesnative*, the default convention provided with the A Series MCP. You then initiate a dialogue with an application to process some documents for your Milan office, and select the Italy convention for that session. When you return to MARC/1, the *Aseriesnative* convention is still active for that dialogue.

You can select a convention for your MARC session by using either MARC menus and screens or the CONVENTION command. You can select a convention for your CANDE session by using the CONVENTION command in CANDE.

The selected convention overrides the default convention for the system and the convention selected at the usercode level for the current MARC session only.

The convention name must be the name of an available convention on the system. To request the names of the available conventions, enter *SHOW CONVENTIONS* in the *Action* field of a MARC screen.

To set the session convention, perform one of the following steps:

- In the *Action* field of a MARC screen, enter

CONVENTION <convention name>

- Enter *GO CONVENTION* in the *Action* field of a MARC screen.
- Access the SC session control menu from the MARC home menu, access the SCONV menu, and then select SCONV.

Setting the Convention at the Usercode Level

The system administrator can use the MAKEUSER utility to associate an individual usercode with a convention other than the system default convention. For information on the MAKEUSER utility, see the *A Series Security Administration Guide*.

Setting the Convention for a Task

An individual task can use the CONVENTION task attribute to specify a convention. This method overrides the currently active convention. However, the specified convention remains active only for the user running that task, and only until the task terminates. See the *A Series Task Attributes Programming Reference Manual* and the *A Series Task Management Programming Guide* for information about the CONVENTION task attribute.

Creating Localized Versions of MARC

Localized versions of MARC are normally produced by Unisys. However, it is possible for you to create a localized version, as discussed in the following paragraphs. For example, suppose that your company has opened a new division in Montreal and you have been asked to provide French versions of the MARC menus you have customized for your division.

Consider the following factors if you intend to create a localized version of MARC:

- Menus and forms
- Online documentation
- Help text keywords
- MARC system messages
- System output

The following paragraphs discuss each of these factors. The discussion assumes that you have already read the *A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide*.

Localizing MARC Menus and Forms

To localize the MARC menus and forms, you must first generate a localized MARC menugraph file. You then position the name of that file at the appropriate location in the active menugraph list.

Generating the Localized MARC Menugraph File

The Interactive Menugraph Generator (IMG) is the tool you use to perform this step. IMG enables you to change the selection keys and selection descriptions displayed on a menu, and to change the descriptive text that is displayed on forms. You can choose

phrases in an alternate language with meanings equivalent to the English text that had been shown on the screens. You perform the actual work of translation; IMG supplies the means of implementing the translated version as a new menugraph file. See the *A Series Interactive Menugraph Generator (IMG) Operations Guide* for detailed information.

Positioning the File in the Active Menugraph List

Once you have created a localized menugraph file, you need to determine what position that file is to occupy in the active menugraph list for your session. The position the file occupies in this list determines at what point MARC accesses the file. See “Using Multiple MARC Menugraphs” in Section 8, “Creating Custom Versions of MARC,” for a discussion of the predefined search order MARC follows to locate the menugraph containing a particular requested screen.

Each of the menugraphs in the active menugraph list can be written in a different language. To position the localized menugraph file as the substitute menugraph – the first menugraph MARC searches for any requested screens – use the following form of the GO action:

```
GO <screen name> IN <menugraph file title>
```

The screen name is the name of any screen in the localized menugraph file; the menugraph file title is the title of that file.

For additional information on the active menugraph list and the order in which MARC searches the menugraph files for a particular screen, see “Active Menugraph List” in Section 8, “Creating Custom Versions of MARC.”

MARC displays the action words – the words listed on line 3 of all MARC screens to indicate the available actions – from the menugraph associated with the current MARC session language. For information on how to change the MARC session language, see “Setting the MARC Session Language” earlier in this section.

If you have named another language as the MARC session language, MARC uses any action words defined in the menugraph associated with that language rather than those defined in the localized menugraph, even if you declare that menugraph as the substitute menugraph. For example, suppose you name Italian as the session language and then use the GO action to access a screen in the French menugraph file, thereby declaring that menugraph as the substitute menugraph. MARC continues to display action words from the Italian menugraph.

Localizing the MARC Online Documentation

The help books for online documentation are linked to individual menugraphs. When you request help on a given screen, MARC displays the text from the help books associated with the menugraph from which the screen was chosen. If the help books associated with that menugraph have been translated, MARC displays the translated text.

Creating Translated Help Text

You can use the MARC help book source files—SYMBOL/MARC/ENGLISH and SYMBOL/MARC/SHORTHELP/ENGLISH—as the starting points for generating your own help books and translating the existing online documentation into another language.

As with menugraph files, you do the actual work of translation. The A Series Editor enables you to enter your translations into help book source files. The Help Utility then uses those help book source files to create help book files containing the help text in a form that MARC can access. For information on the Editor, see the *A Series Editor Operations Guide*. For information on the Help Utility, see the *A Series Help Utility Operations Guide*.

Associating the Translated Text with the Localized Menugraph File

IMG is the tool you use to associate the translated help book files with a localized menugraph file so that they become the default books for that menugraph. See the *A Series Interactive Menugraph Generator (IMG) Operations Guide* for more information.

Localizing Help Text Keywords

Keywords are the means by which MARC associates online documentation with a particular portion of a particular screen. Keywords appear in the menugraph, and corresponding keywords appear in the help book source file. Unisys recommends that you do not translate the keywords that are common to the menugraph and help book source file, since most of these keywords are cryptic and users do not normally see them. If you do translate the menugraph keywords, however, be sure to translate the corresponding keywords in the help book source file as well.

The help book source files also contain certain keywords embedded in the help text itself, and you normally translate these keywords as part of the text. An embedded keyword is a link to a related piece of additional help text.

For example, recall the long help text message illustrated in Figure 3-2 for the AX selection of the JDC menu. This message includes **mix numbers** to indicate that an online definition of the term is available. If you translate an embedded keyword, take care to translate every occurrence of that keyword in exactly the same way.

Within the help book source file, each keyword is preceded by a .KEYWORD command and followed by the help text related to that keyword. For example, the help text illustrated in Figure 3-3 is the result of the following portion of the help book source file SYMBOL/MARC/ENGLISH:

```
.SECTION "MIX NUMBER"  
.KEYWORD MIXNUMBER *MIX NUMBER* *MIX NUMBERS*  
A mix number is the four-digit number assigned by the system to identify  
each job or task on the system.
```

It is this occurrence of a given keyword in particular – its appearance after the .KEYWORD command – that links to the additional help text for that keyword. Be especially careful to change this occurrence of the keyword, and all other occurrences of the keyword in the help book source file. They must all match exactly to provide access to all the available help text in the file.

Any help text can contain embedded keywords. Users can select as many embedded keywords as they want. Embedded keywords enable one piece of help text to link to another piece, which in turn can link to another, and so on. To preserve this linkage, make sure that all occurrences of any keyword you translate are translated consistently throughout the help book source file. For detailed information on the .KEYWORD command, see the *A Series Help Utility Operations Guide*.

Localizing MARC Messages

Most messages that MARC displays are found in the MARC software. The messages provided with MARC are in English, but you can create new messages in another language by using the Message Translation Utility (MSGTRANS).

MSGTRANS can assist you in translating messages and also binds the localized messages to the MARC software, making a new version of MARC available. To use MSGTRANS to translate messages to another language, you must add the name of the new language to message 0 of the MLSMSG array in the file *SYSTEM/MARC/COMMANDER, the file that is the main body of MARC. The name of every language used by MARC, except English, must be included in message 0 of this array.

The following steps briefly explain how to use MSGTRANS to add a new language to the MLSMSG array, translate messages, and bind the new language to the MARC software. For a detailed discussion of MSGTRANS, see the *A Series Message Translation Utility (MSGTRANS) Operations Guide*.

Note: *Because the MSGTRANS utility creates a new version of the code file SYSTEM/MARC/COMMANDER, it is good practice to save a copy of the current version of this file before you access MSGTRANS.*

The steps to use MSGTRANS to add a new language are as follows:

1. Take one of the following actions to access the MSGTRANS utility:
 - Choose TOOLS from the MARC home menu and transmit, and then select the MT keyword from the TOOLS menu and transmit.
 - In any CANDE session, transmit

```
RUN $SYSTEM/MSGTRANS ON <family name>
```

- From the *Action* field of any MARC screen, enter and transmit

```
RUN SYSTEM/MSGTRANS ON <family name>
```

If you use the RUN command, use the syntax *ON <family name>* to identify the disk family on which the program resides. If the program is on your default family, you can omit this syntax from the statement.

The utility displays the start session screen. Figure 9-1 illustrates this screen.

```
What object file is to be edited? [      ]  
  
To exit the program, enter ABORT here. [      ]  
  
-----  
  
HELP      For help at any time, press the SPCFY key. Information will be  
          displayed about the field or caption where the cursor is  
          located.  
  
REFRESH   To refresh the screen at any time, type the following sequence:  
          CTRL Ø Ø XMT  
  
-----  
  
Unisys Message Translation Utility, Version 39.042.040
```

Figure 9-1. MSGTRANS Start Session Screen

2. Enter **SYSTEM/MARC/COMMANDER* in the *What Object File Is To Be Edited?* field of the start session screen and transmit.

MSGTRANS then displays the MSGTRANS home screen, as illustrated in Figure 9-2.

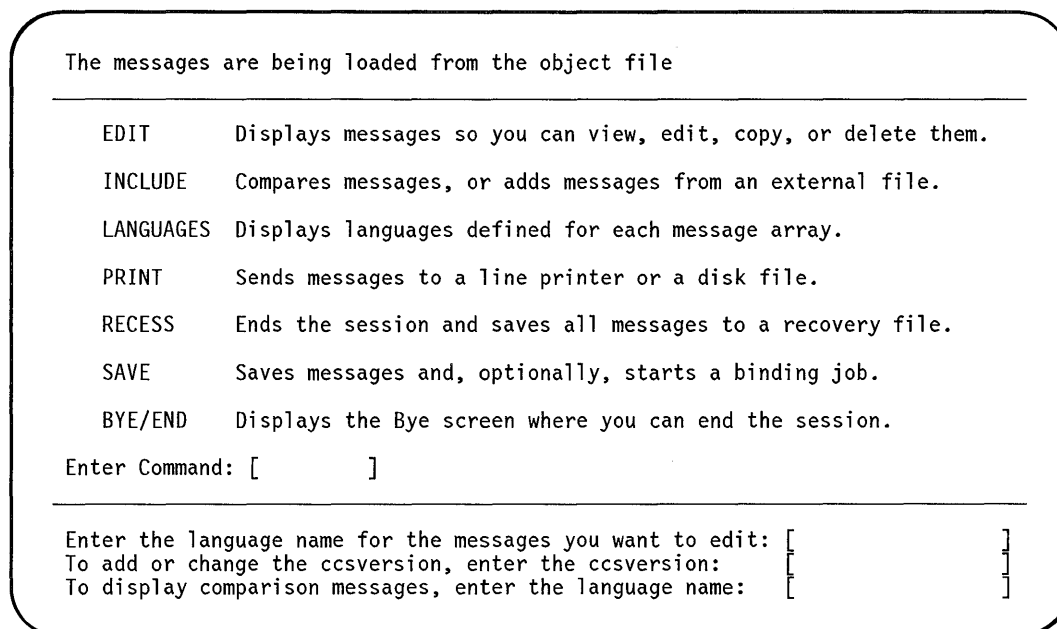


Figure 9-2. MSGTRANS Home Screen

3. Enter *EDIT* in the *Enter Command* field, and move the cursor to the next field.
4. Enter the name of the language you want to add in the *Enter the language name for the messages you want to edit* field – for example, *FRANCAIS*. Move the cursor to the next field.
5. Enter *ENGLISH* in the *To display comparison messages, enter the language name* field, and transmit.
MSGTRANS displays the Edit screen, showing the first physical message in the file **SYSTEM/MARC/COMMANDER*.
6. Enter *FIRST OF MLSMSGS* in the *Where Next* field and transmit.
MSGTRANS refreshes the Edit screen and displays message 0.
7. Translate the word *OK* into the desired language. The translated word must be enclosed in quotation marks. For example, you could transmit "*Bon*" to translate the word into French.
8. The next step is to translate any or all MARC messages into the new language – see the *A Series Message Translation Utility (MSGTRANS) Operations Guide* for details. However, translating the contents of message 0 is the only step that is *required* to bind a language to MARC.
9. Enter *SAVE* in the *Where Next* field.
MSGTRANS displays the MSGTRANS Save screen.
10. Bypass the *Select OBJECT or SOURCE* field. MSGTRANS prefills this field with the word *OBJECT*.

11. Enter a name in the *Enter File Title . . .* field and transmit. This name is the title you want to give to the saved messages file. For example, you might name a French message file FRANCAISMSGs. Move the cursor to the next field.
12. Bypass the *What Languages Are to Be Saved?* field.
13. Enter **SYSTEM/MARC/COMMANDER* in the *To Bind the Saved Messages . . .* field and transmit. MSGTRANS displays the MSGTRANS restart screen.
14. From the *Enter Command* field, initiate one of the following actions:
 - Enter *END* to exit MSGTRANS.
 - Enter *HOME* to establish another language. Repeat all steps just outlined, beginning with step 2.

Storing Localized Versions of Other MARC Screen Elements

The following elements of MARC screens are also stored as messages and can be translated and stored using MSGTRANS:

- The prompt that appears before the *Action* field on line 2.
- The “Press SPCFY for Help” and “SPCFY again for more” messages that are displayed on line 3.

Appendix A

Handling Faults in MARC

What Happens at the ODT

When MARC encounters a fault, the system displays the following message at the ODT and waits for the operator to enter an AX (Accept) or DS (Discontinue) system command:

```
A MARC STACK (MIX # <mix>) HAS JUST FAULTED.  
EXAMINE SUMLOG FOR EVIDENCE OF FAULT.  
PLEASE SUBMIT UCF, INCLUDING (DUMP) FILE: <dump file name>.  
(PRINTED OR ON TAPE) AND SUMLOG (ON TAPE) COVERING TIME OF FAULT.  
>>>  MARC FAULT - WAITING FOR AX/DS  <<<.
```

The effect this situation has on MARC processing depends on how MARC is configured, as follows:

- If the system is configured to run with only one MARC worker stack in the mix, MARC stops all activity until the operator enters a DS (Discontinue) or AX (Accept) system command that specifies the mix number listed in the message.
- If MARC is configured to run with multiple MARC worker stacks in the mix, only the stack listed in the message is halted; other MARC worker stacks continue to function normally.

In this situation, the operator is advised to enter an AX or DS command for any waiting MARC worker stack since a stack in the waiting state remains in that state indefinitely until it is discontinued; the system does not initiate a new stack to replace a waiting stack.

If a DS command – entered either by an operator or through the DCKEYIN function in a privileged Data Communications ALGOL (DCALGOL) program – instructs the system to discontinue a MARC worker stack that has not faulted, the following messages appear at the ODT, and MARC does not wait for an AX command:

```
A MARC STACK (MIX # <mix>) HAS JUST FAULTED.  
CAUSE OF FAULT: OPERATOR DS
```

If a dump file exists, the following message also appears:

```
DUMP FILE TITLE: <dump file name>.
```

If no dump file exists, the following message appears:

```
NO DUMP FILE WAS CREATED.
```

What Happens at Your Terminal

When a fault condition occurs or MARC is discontinued, MARC recognizes that a worker stack is terminating. Subsequent events depend on the current situation, as follows:

- If the stack has been initialized and work is in progress for a user dialogue, MARC attempts to continue processing from the point at which the fault occurred or as close to that location as possible. MARC displays the home menu at the user terminal where input was being processed when the fault occurred.

If the user is in the process of logging on, the system either continues the logon sequence from where it left off or repeats the entire logon sequence.

- If the user has already entered *BYE* and logged off, recovery might or might not be necessary for the dialogue—and therefore a message might or might not appear on the terminal screen—depending on where the fault occurred, as follows:
 - If the dialogue is not logged in at the time the fault occurs and it is necessary for MARC to recover the dialogue, the following message appears at the bottom of the terminal screen:

```
A fault has occurred during logon or after logoff. Please
try again. If you continue to have problems, notify your
supervisor or Operations.
```
 - If the dialogue is logged in when the fault occurs, MARC displays the following message:

```
A fault has occurred in processing one of your
commands. Please notify your supervisor or Operations.
You may have to re-enter some commands to continue.
```

If you are running MARC from a Teletype-like® terminal, this same functionality applies—that is, MARC attempts to recover from the situation. Error messages appear to inform you of the fault, and the system attempts to return to the state that existed at the time of the fault. However, the system sends no screens to Teletype-like terminals.

What You Can Do

After the message announcing the fault condition appears, you can continue working without interruption; however, you might not be able to continue where you left off. You might need to reenter the command that was being processed when the fault occurred. Moreover, you must reenter any commands that have not yet been processed. For example, if you are in the middle of a multistep request or chain of commands, or if you have queued some commands to MARC, you must reenter those commands.

After you reenter any necessary commands, MARC is in a proper state to accept another message from you, and is unlikely to encounter other faults.

Teletype is a registered trademark of Teletype Corporation.

If you continue to experience problems or continue to receive messages indicating that more faults have occurred, notify a supervisor or your operations staff.

In some instances, the system might fail to display a home menu or notification message after a MARC fault. Because of the way it processes some MARC and COMS commands, the system is sometimes unable to send this screen or message to the appropriate dialogue and terminal.

In these cases, a continuation command (+) might be accepted by MARC after the fault. If the continuation command is unsuccessful, try reentering the last command. If MARC fails to honor the request, begin the current command sequence again. If MARC still rejects the request, notify a supervisor or operator so he or she can perform a system memory dump and submit a User Communication Form (UCF) to Unisys.

Appendix B

Installing MARC on Your System

This appendix is intended for system administrators. It describes the steps you must take to ensure that a fully operational version of MARC is running on your system:

1. Install the MARC software.
2. Configure the ODT for data comm mode.
3. Configure the user terminals.
4. Initialize MARC.

The preceding steps form part of the procedure for setting up your system. This procedure is described in more detail in the installation guide specific to your model of A Series system – for example, the *A Series A 1–A 6 Systems Software Installation Guide*. This appendix provides information about these steps that is specifically related to MARC.

If your site is licensed to receive source code files, you can optionally recompile the MARC software before installing it, in order to include one or more compile-time options. The MARC compile-time options are described in the *A Series Software Release Installation Guide*.

This appendix also discusses how MARC can process messages produced by your installation's billing utility.

Installing the MARC Software

The MARC software is installed during installation of a Mark release or a support release, as described in the *A Series Software Release Installation Guide*. For MARC to be operational, COMS must also be installed.

You can also choose to install the MARC software independently of a Mark release or support release installation. The MARC software is contained in several files. These files must all be loaded on the same family and under the same usercode. Typically, system files are installed under the nonusercode usercode, an asterisk (*). However, you can install the files under a different usercode if you want to restrict access to MARC. Each of the following files must be available for MARC to run with full capabilities:

Installing MARC on Your System

File Name	Function
SYSTEM/MARC/COMMANDER	This file is the main body of MARC. It contains the code that processes menus, forms, and commands.
SYSTEM/MARC/AGENDA/TDXXX	This file contains the COMS agenda used to communicate with TD-like terminals.
SYSTEM/HELP	This is the code file for the run-time help library.
SYSTEM/MARC/MENUGRAPH/ENGLISH	This file contains the menugraph that is released with MARC.
BOOK/MARC/ENGLISH	This file contains the long help text displayed by MARC.
BOOK/MARC/SHORTHELP/ENGLISH	This file contains the short help text displayed by MARC.

Files You May Omit

If you omit SYSTEM/HELP and the BOOK files from the installation, MARC runs, but users are unable to access the online help information in MARC. You can omit SYSTEM/MARC/MENUGRAPH/ENGLISH, provided that another menugraph file is made available.

System Libraries

SYSTEM/MARC/COMMANDER must be defined as a system library by the name of MARCSUPPORT. To make SYSTEM/MARC/COMMANDER a system library, use the SL (Support Library) system command as follows:

```
SL MARCSUPPORT = *SYSTEM/MARC/COMMANDER
```

SYSTEM/HELP can also be a system library. Although MARC can use SYSTEM/HELP even if it is not a system library, you must make SYSTEM/HELP a system library by the name of HELPSUPPORT for other products to be able to use it. To make SYSTEM/HELP a system library, use the SL (Support Library) command as follows:

```
SL HELPSUPPORT = *SYSTEM/HELP
```

See the *A Series System Commands Operations Reference Manual* for a description of the SL command.

Configuring the ODT for Data Comm Mode

Two modes are available at every operator display terminal (ODT) that can be configured as a data comm terminal: system command mode and data comm mode. In *system command mode*, the ODT responds to system commands as described in *A Series System Commands Operations Reference Manual*. In *data comm mode*, all the features of COMS are available at the ODT so that the MARC window is available for running MARC, as are the other windows that COMS provides. MARC has the same

capabilities when run at an ODT as it does when run at a remote terminal, plus the following additional features:

- You can use the ??MEMDP (Dump Memory) primitive command to cause a memory dump. This command is described in the *A Series System Commands Operations Reference Manual*.
- If system security permits, you can use an asterisk (*) instead of a usercode. For more information, see “Super User” in Section 7, “Controlling System Security.”

Requirements for Data Comm Mode

An ODT can run in data comm mode if it can be configured as follows:

- The terminal must be set to suppress the end-of-text character (ETX) that is normally appended by the ODT to each line of printed output.
- The terminal must be set to interpret any internal DC1 messages received from the system as meaning that the ODT should remain in the receive mode after transmission. (ODTs are normally configured to respond to a DC1 message by erasing the remainder of the current line.)
- The default placement of the cursor must be set to the upper left-hand corner of the screen.
- The terminal must be set to transmit a full screen at a time.

Using Two Modes on Your System

If you have multiple ODTs attached to your system, you can run some in data comm mode and some in system command mode.

By leaving at least one ODT in system command mode, you have continual access to *automatic display mode*, which is set by the ADM (Automatic Display Mode) system command, and to all the primitive system commands. Automatic display mode provides a continually updated display of system status information.

The ODTs running in data comm mode provide you with access to the MARC system, which enables you to perform system functions without knowing the command syntax. In data comm mode, you also have access to the CANDE window and other available COMS windows, so you can use the ODT as a remote terminal.

Caution

Programs that open files with the KIND attribute equal to SPO might run erratically at an ODT that is configured like a data comm terminal while it is in system command mode.

Switching between Modes

If you have an ET 2000 ODT that is attached to an A 9 system, you can reconfigure that terminal to data comm mode by pressing, in order:

1. CTRL (control) key
2. Character key *C*
3. XMIT (transmit) key

To return the terminal to system command mode, repeat the key sequence. On an A 9 system, this key sequence acts as a toggle switch to reconfigure the terminal. Otherwise, see the reference manual appropriate to your terminal type for information about reconfiguring your ODT.

In addition, you can use the TERM (Terminal) system command with the DCSTATION option set to TRUE if you plan to switch back and forth between data comm mode and system command mode at a given terminal. This option causes the NS (Next Screen) prompt to appear at the upper left-hand corner of the terminal instead of the lower right-hand corner. The TERM command is described in the *A Series System Commands Operations Reference Manual*.

Limitations

The data comm mode for an ODT has been qualified only for applications run through CANDE and MARC. Those message control systems (MCSs) that use the set application number DCWRITE (type = 38) are not supported at the ODT. Any attempt by an MCS to perform that DCWRITE for the ODT is ignored.

Commands for Data Comm Mode

The following system commands are used for initiating or ending data comm mode, or for displaying information about the ODTs in data comm mode. For detailed descriptions of these commands, see the *A Series System Commands Operations Reference Manual*.

??MARC (Transfer to Menu-Assisted Resource Control) Primitive Command

Entering the ??MARC primitive command initiates data comm mode at the ODT by causing the system to direct control of the ODT to the COMS/ODT/DRIVER program. COMS/ODT/DRIVER automatically releases the ODT to COMS, and usually causes the LOGON screen for MARC to be displayed. COMS/ODT/DRIVER must be declared as an MCS.

??ODT (Return from Menu-Assisted Resource Control) Primitive Command

Entering the ??ODT primitive command forces an ODT to leave the control of the COMS/ODT/DRIVER program and reenter *system command mode*, returning the

processing of subsequent input to the operating system. This command takes effect only if the ODT is currently in data comm mode. Any task running under the MARC interface continues running. You can return to these tasks by entering `??MARC`.

SM (Send to MCS) System Command

You can use the SM (Send to MCS) system command to send any of the following three control commands to the COMS/ODT/DRIVER program – not to SYSTEM/COMS – on an ODT that is running in data comm mode through COMS/ODT/DRIVER.

- CLEAR <ODT unit number> control command

The CLEAR control command causes the specified ODT to be disconnected from COMS/ODT/DRIVER. The ODT unit number is included after the word CLEAR in this command. You can use this command to disconnect only ODTs that are currently running in data comm mode. You must then enter the `??ODT` primitive command at the ODT to resume operations. You can return the ODT to data comm mode at any time by using the `??MARC` primitive command.

- ODTS control command

The ODTS control command displays the following information:

- A list of the pseudostation numbers of all the ODTs running in data comm mode, and the corresponding ODT unit number for each.
- A list of all the ODT unit numbers and the corresponding pseudostations for those ODTs that are running in data comm mode.

- QUIT control command

The QUIT control command disconnects all ODTs running in data comm mode from COMS/ODT/DRIVER and takes COMS/ODT/DRIVER out of the mix. You need not enter `??ODT` to return the ODT to system command mode.

Configuring User Terminals

MARC operates from Unisys TD-like terminals, such as the TD830, MT983, MT987, ET 1100, or SR100, as well as from any workstation that can run ET 1100 or MT emulation, such as B 2x, B 3x, and IBM® personal computers (PCs) or compatible PCs.

The TD-like terminals listed in the preceding paragraph operate in a mode known as *forms mode*, in which they are capable of displaying MARC menus and forms.

MARC also provides some capabilities of older terminals that have the line-at-a-time characteristics of a Teletype terminal. Since MARC cannot use menus and forms with these terminals, MARC interacts with them in *single-line mode*, in which the terminal displays and transmits a single line at a time.

Installing MARC on Your System

If your terminal is a Unisys personal computer (PC) or a compatible PC that supports the PCMARC user interface, see the *A Series PCMARC Installation and Operations Guide* for information on the use of PCMARC.

Configuring Terminals in Forms Mode or Single-Line Mode

Most Unisys terminals operate in forms mode. Unless a terminal is one of the older terminals that have the line-at-a-time characteristics of a Teletype, it should be configured in forms mode for MARC operations.

You can use the Interactive Datacomm Configurator (IDC) to configure terminals. The `TERMINAL NAME` attribute in the Network Definition Language II (NDLII) specifies whether a given station is initially in forms mode or single-line mode. If the terminal name in the `SYSTEM/DATACOMINFO` file begins with `TD` or `MT`, then the station is initially configured in forms mode. Otherwise, it is initially configured in single-line mode.

When a given terminal is in single-line mode, you can log on, and, if your usercode has the appropriate privileges, you can enter COMS control commands—those commands that begin with a question mark (?). The `?ON CANDE` command can then be used to release the station to `SYSTEM/CANDE`, which enables the use of all `CANDE` commands at the terminal. At such a terminal, `?WRU` is the only command the system accepts before a user logs on.

Many of the commands discussed in this guide are shown as beginning with a question mark (?) character. This character is defined in NDLII as the station control character for your station. The default station control character is a question mark (?).

Configuring Terminals for Full-Screen Transmission

To promote ease of use when interacting with MARC, configure your terminals for full-screen transmission. See the reference manual for the type of terminal you have to determine how to configure a terminal to full-screen transmission.

Full-screen transmission means that all information in all input fields on the screen is transmitted, regardless of the position of the cursor, when you press the `XMIT` key. Be aware, however, that any input in the *Action* field of a screen overrides input in all other fields on the screen and causes the system to ignore any input in other fields. If the *Action* field is left blank, any entries in the other input fields on the screen are accepted.

If a terminal is not configured for full-screen transmission, the position of the cursor at the time you transmit becomes important. You can still transmit all the information you have entered in input fields on the screen by pressing the `HOME` key before you transmit. Pressing the `HOME` key moves the cursor to the start of the *Action* field. If the cursor is not located at the start of the *Action* field when you transmit, only information on lines above the cursor, and on the same line as the cursor but to the left of it, is transmitted. As a result, if you leave your cursor in the middle of the information you are trying to transmit, some of the information is lost.

If your terminal is not configured for full-screen transmission, Unisys recommends that you place the cursor at one of the following positions before you transmit:

- At the start of the *Action* field
- After the last field in which you have entered information

Any prefilled default values that are displayed in input fields are used even if they are not transmitted, unless other values are transmitted in their places.

Initializing MARC

The installation guide for the particular model of A Series system you are using provides information on installing software and initializing your system. To initialize MARC, use the primitive command ??MARC (Transfer to Menu-Assisted Resource Control), following the instructions in the appropriate installation guide for your system.

Reviewing Messages Produced by Your Billing Utility

If your installation has used the interfaces and functions provided by Unisys to build a billing utility, MARC can supply information to, and display information returned from, that utility. For example, MARC can request that bill processing be performed whenever any of the events listed in Table B-1 occur. The second column indicates whether or not MARC displays billing messages for a particular event.

Table B-1. Sample Billing Utility Bill Processing Schedule

Event	Billing Messages Displayed
Beginning of session (BOS)	None
End of session (EOS) initiated by HELLO command	All
End of session (EOS) initiated by SPLIT command	All
End of session (EOS) initiated by BYE command	None
End of session (EOS) initiated by COMS command CLOSE	None
Beginning of task (BOT)	All
End of task (EOT)	All
Session or task status inquiry	All

For detailed information on the A Series billing support facilities, see the *A Series System Administration Guide*.

Appendix C

Commands Available in MARC

This appendix lists all the commands that are available in MARC. The commands are grouped according to whether they are MARC commands, Communications Management System (COMS) commands, system commands, Print System commands, or Work Flow Language (WFL) statements entered as commands. Within each group, the commands are listed in alphabetical order.

MARC Commands

The commands listed in Table C-1 are MARC commands even though some of them are similar to or the same as CANDE commands (for example, FAMILY, FILES, LIST, NEWS, and WRU).

Table C-1. MARC Commands

Command	Function
ALLMSG	Displays recent system messages
BDIR	Displays the directory of your printer backup files
BP	Suspends the program being tested under Test and Debug System (TADS)
DIRECTIVE	Permits creation of user-defined commands
FAMILY	Displays or sets disk family specifications
FILES	Displays file titles (same as PD)
GENERATE PASSWORD	Generates and displays the system-generated potential password
HELP	Displays help text for a keyword (same as TEACH)
LANGUAGE	Displays or sets MARC and COMS application session languages
LIST	Lists all or part of a file
MENU	Displays active menugraphs for the current session
MSC	Suppresses display of system messages according to content and destination
MSG	Displays your task messages for the current session
NEW	Updates the menugraph or help book in use

continued

Commands Available in MARC

Table C-1. MARC Commands (cont.)

Command	Function
NEWS	Displays the news file
PD	Displays file titles (same as FILES)
PDIR	Displays information about file lists
SHOW PASSWORD	Displays the system-generated potential password
SMSG	Displays the last 20 (approximately) system messages
SUSPENDUSERCODE	Suspends use of your usercode
TEACH	Displays help text for a keyword (same as HELP)
WRU	Displays station status

COMS Commands

All the COMS commands, listed in Table C-2, are available through MARC.

Table C-2. COMS Commands Available in MARC

Command	Function
ADDSTA	Adds a station to a line
ATTACH	Achieves COMS control of station
BYE	Ends a COMS session
CLOSE	Closes a COMS window or dialogue
COMS	Controls the transaction trail for a nondatabase program
CONNECT	Transfers the station to a remote host system
DATABASE	Controls the transaction trail for a database
DISABLE	Disables a station, window, database, or library
ENABLE	Enables a station, window, database, or library
HELLO	Starts a new COMS session
JOBS	Displays a list of programs that are running
MCS	Transfers control of any station to another MCS
MONITOR	Monitors COMS activities
MOVE	Moves a station to another line

continued

Table C-2. COMS Commands Available in MARC (cont.)

Command	Function
ON	Switches to another COMS window
PASS	Sends text to another window dialogue
PASSWORD	Alters the password for a particular usercode
PURGE	Deletes messages temporarily stored for a station
QUIT COMS	Shuts down the COMS network
READY	Makes a station ready to be polled for input
RECALL	Retrieves messages for a station, and delivers them to another station
RELEASE	Transfers control of a station to another MCS
REPORT	Displays COMS network-related information
RESUME	Resumes suspended display of messages from window dialogues
SAVE	Makes a station not ready to be polled for input
STATUS	Displays the status of a station, window, program, or database
SUBTRACT	Removes a station from the line
SUSPEND	Suspends the display of messages from window dialogues
SWAP	Interchanges two lines
TO	Sends a message to another station or to the ODT
TRACE	Records COMS diagnostic data
WINDOWS	Lists current COMS windows

System Commands

The system commands listed in Tables C-3 and C-4 are available through MARC. The MSG system command is available in MARC, but only through the SMSG syntax.

Table C-3. System Commands Available in MARC

Command	Function
A	Lists all active jobs and tasks
ACCOUNTING	Sets attribute values for resource accounting
ACQUIRE	Acquires a resource for the active configuration group

continued

Commands Available in MARC

Table C-3. System Commands Available in MARC (cont.)

Command	Function
AD	Duplicates or deletes the active disk access structure
AR	Releases the old archive log and creates a new one
ARCCOPY	Makes a safety backup of the archive directory
ARCDUPLICATE	Makes a duplicate copy of the archive directory
ARCREPLACE	Replaces an old or bad archive directory
ASD	Displays information about the actual segment descriptor (ASD) table
ASDU	Displays information about ASD usage
AT	Sends a system command to another network host
AUTORESTORE	Copies a file from tape if it is unavailable on disk
AX	Passes text to a program
BNA	Starts or stops a portion of BNA Version 2
BNAVERSION	Sets or displays the version of BNA to be used after a halt/load
BR	Starts a checkpoint for a task
C	Lists recently completed jobs and tasks
CA	Displays unconnected network subports
CF	Sets or displays the title of the system configuration file
CL	Clears exception condition flags for a peripheral
CLOSE	Logically detaches a disk unit from the system
CM	Sets or displays the version of the operating system to be used
CO	Sets or displays compiler options
COMPILERTARGET	Sets the default target value for compilers
COPYCAT	Creates an inactive copy of the system catalog
CP	Designates a code file as a control program
CQ	Clears the system messages queued for display
CS	Designates a code file as the supervisor program
CU	Displays information about memory usage

continued

Table C-3. System Commands Available in MARC (cont.)

Command	Function
DA	Runs the DUMPANALYZER utility to analyze a memory dump
DBS	Displays information about database stacks
DD	Creates or deletes the backup copy of a flat directory
DF	Empties the dump from a dumpdisk file
DIR	Runs the FILEDATA utility to display a disk directory
DL	Indicates the name of the disk family containing the system files
DN	Creates a disk file for system memory dumps
DO	Sets or displays diagnostic options (A 12, A 15, A 17)
DQ	Designates a job queue as the system default queue
DR	Changes the system date
DRC	Activates disk resource control
DS	Terminates a job or task
DU	Displays the amount of available disk space
DUMP	Performs a memory dump
FA	Changes file attributes
FAS	Displays memory errors (A 12, A 15, A 17)
FM	Assigns the output device for a file with the FORMID attribute
FR	Designates a tape reel as the final reel of a file
FREE	Logically detaches a resource from a configuration group
FS	Forces the start of a scheduled job or task
GC	Creates input for the system configuration file
HI	Causes the exception event for task
HLUNIT	Specifies a halt/load unit
HOSTNAME	Sets or displays a network hostname
HS	Controls the scheduling of jobs and tasks
HU	Sets a usercode for BNA Host Services requests
IB	Displays operator instructions for a job

continued

Commands Available in MARC

Table C-3. System Commands Available in MARC (cont.)

Command	Function
ID	Initializes data communications
IK	Installs a keys file for release tapes
IL	Ignores the label for a file
INSTALL	Installs the system software
J	Lists the active jobs and tasks by job structure
LB	Changes the label of a disk or host control (HC) unit
LC	Enters a comment in the system log file
LD	Loads the control records to or from a tape file
LG	Logs events for individual tasks (InfoGuard only)
LH	Loads the control firmware to a disk controller
LIBS	Displays information about libraries
LJ	Enters a comment for a job or task in the system and job logs
LOG	Runs the LOGANALYZER utility to analyze the system log file
LOGGING	Sets or displays the system logging options
LP	Prevents program termination by DS or QT commands
MA	Enables or disables APL access to a file
MB	Designates a file as a BOOTCODE file (A 1 through A 6)
MC	Designates a file as a compiler code file
MCS	Displays information about message control systems (MCSs)
MDT	Sets or displays the memory dump type
MIRROR	Controls Mirrored Disk usage
ML	Sets or displays the current mix limit
MM	Displays information on memory modules
MODE	Controls the status of a peripheral
MOVE	Moves jobs in a queue or moves a disk to another drive
MP	Marks a program with a system message identifier
MQ	Creates, modifies, or deletes a job queue

continued

Table C-3. System Commands Available in MARC (cont.)

Command	Function
MSG	Displays recent system messages (only through SMSG syntax in MARC)
MU	Creates a new usercode and password
MX	Lists the active jobs and tasks with messages
NA	Routes BNA commands to BNA
NC	Sends network control commands to BNA Version 2
NET	Starts or stops BNA Version 1
NETEX	Starts or stops NETEX® software†
NF	Resolves a no-file condition for a COBOL85 program
NOTOK	Does not reactivate a suspended task
NS	Displays the next screen of command output
NW	Sends an operator command to BNA Version 1 or 2
NX	Sends operator input to a NETEX library
OF	Resolves a no-file condition for an optional file
OG	Sets or displays the overlay goal factor for tasks
OK	Reactivates a suspended task
OL	Displays label and path information for a peripheral
OP	Sets or displays operating system options
OT	Displays the stack cell for a task
OU	Sends task output to a peripheral
PA	Associates input devices
PB	Prints or punches backup disk or tape files
PD	Displays a list of the files in a disk directory
PER	Displays the status of a peripheral device
PF	Displays a FETCH message of a queued WFL job
PG	Purges a tape, disk, or host control (HC) unit
PLI	Sets the interval for logging data on open files
POWER	Powers a system or disk drive up or down

† NETEX is a registered trademark of Network Systems Corporation.

continued

Commands Available in MARC

Table C-3. System Commands Available in MARC (cont.)

Command	Function
PP	Designates a code file as a privileged program
PQ	Removes all jobs from a job queue
PR	Sets the priority of a job or task
PS	Sends a command to the Print System
PV	Displays volume library information for a disk or tape
QF	Displays information about a job queue
QT	Terminates the output of a printer backup file
RB	Rebuilds the access structure for the flat directory of a disk
RC	Reinitializes a disk volume
RECONFIGURE	Reconfigures a system according to a new configuration
REMOTESPO	Activates a user terminal as a remote ODT station
REPLACE	Makes an exact duplicate of a disk volume
RES	Marks an area of disk space as unusable
RESTART	Enables a job to restart without a halt/load
RESTRICT	Sets security restrictions
RF	Displays the reliability factor for a device
RM	Removes the file specified in a DUP LIBRARY message
RP	Designates a code file as a resident program
RW	Rewinds, unloads, and locks a tape drive
RY	Makes a device ready if it was made unready by another command
S	Displays the scheduled jobs and tasks
SB	Substitutes one backup medium for another
SBP	Sets the interval for computing system usage information
SC	Displays the current system configuration
SCAN	Checks a disk volume for read errors
SECOPT	Sets or displays the security options (InfoGuard only)
SEGARRAYSTART	Sets or displays the point after which arrays are segmented
SEND	Sends a message to an image printer or network processor

continued

Table C-3. System Commands Available in MARC (cont.)

Command	Function
SF	Sets or displays memory management factors
SHOWOPEN	Lists the names of files that are still open on a disk
SI	Designates a file as a system-intrinsics file
SL	Associates a function name with a library code file
SM	Sends a command to a database or message control system (MCS)
SN	Reinitializes a tape volume
SQ	Displays information about the jobs in a job queue
SQUASH	Consolidates space on a disk family
SR	Ensures that card reader input contains a USER statement
SS	Sends a message from an ODT to a user terminal
ST	Suspends a job or task temporarily
STARTTIME	Sets the start date and time for a queued job
STRUCTURECACHE	Sets caching for memory structures
SUPPRESS	Suppresses the display of an active job or task
SUPPRESSWARNING	Displays or changes the list of suppressed warning messages
SV	Makes a device inaccessible to the system
SYSTEMLANGUAGE	Sets the default language for system messages
SYSOPS	Sets the system options for international conventions
TD	Displays the system time and date
TDIR	Runs the FILEDATA utility to list a tape directory
THAW	Changes the status of a library from permanent to temporary
TI	Displays the time accumulator values for a task
TL	Starts a new system log file and releases the old one
TR	Sets the system date and time
TV	Displays information about a tape volume
U	Displays information about system usage
UL	Resolves a no-file condition by assigning an unlabeled file

continued

Commands Available in MARC

Table C-3. System Commands Available in MARC (cont.)

Command	Function
UQ	Associates an input device with a job queue
UR	Makes a device inaccessible for maintenance purposes
W	Displays the waiting jobs and tasks
WM	Displays version information about the operating system
XD	Makes defective disk sectors unavailable for use
XP	Designates a code file as an executable program
X25MCS	Starts or stops an X.25 MCS
Y	Displays status information about a job or task

The primitive system commands listed in Table C-4 are available at ODT stations in connection with MARC.

Table C-4. Primitive Commands Available in MARC

Command	Function
??MARC (Transfer to Menu-Assisted Resource Control)	Switches a station to data comm mode
??MEMDP (Dump Memory)	Initiates a nonfatal memory dump
??ODT (Return from Menu-Assisted Resource Control)	Returns a station to ODT mode

Print System Commands

All the Print System commands, listed in Table C-5, are available through MARC.

Table C-5. Print System Commands Available in MARC

Command	Function
PS ADDFILES	Creates print requests for existing backup files that have no active tasks or print requests associated with them
PS ASSOCIATE	Associates a usercode or input device with an output device or group, or displays a list of these associations

continued

Table C-5. Print System Commands Available in MARC (cont.)

Command	Function
PS BANNER	Defines or displays the format of the banner page
PS BNA	Enables or disables BNA print routing or its trace feature, or indicates whether BNA print routing is enabled
PS CONFIGURE	Assigns characteristics such as form type and line limit to specified printers
PS DEFAULT	Specifies or displays the type of printer used to print backup files and job summaries, or specifies or displays the default print disposition, which determines the point in a job at which a print request is generated for a file
PS DELETE	Deletes print requests from the print queue
PS DEVICES	Adds a printer to, or removes it from, the pool of printers available for print requests that have no destination specified, and displays characteristics of all system printers
PS FORCE	Prints a specified request as soon as possible
PS GROUP	Displays the names of device groups known to the system and the devices belonging to each group, and creates a device group
PS HEADER	Defines the format of the header page or displays the current header format description
PS MODIFY	Changes the characteristics of a print request
PS NOTOK	Causes the forms alignment process to be repeated
PS OK	Resumes printing of a request that has been stopped by the PS STOP command
PS QUIT	Stops the Print System
PS REMSERVERS	Controls printing of requests at remote printers
PS REQUEUE	Stops printing the request on a specified device, and returns the request to the print queue, marked as an exception, to await further operator action
PS RESTART	Restarts the Print System after a PS QUIT command
PS SELECTION	Sets or displays the print request selection criteria used by the system to determine the device on which a request is printed
PS SERVERS	Specifies the maximum number of print servers available for printing backup files
PS SHOWREQUESTS	Displays information about print requests
PS SKIP	Causes the output printing on a device to skip a specified number of lines or pages, or to skip to the end of the file

continued

Commands Available in MARC

Table C-5. Print System Commands Available in MARC (cont.)

Command	Function
PS STATUS	Displays the status of the printing devices connected to the system
PS STOP	Stops printing on a specified device
PS TRAILER	Defines the format of the trailer page or displays the current trailer format description

WFL Statements Entered as Commands

All the WFL statements, listed in Table C-6, are available through MARC. The characters *WFL* must precede all WFL statements except ADD, CHANGE, COPY, REMOVE, RUN, SECURITY, and START.

Table C-6. WFL Statements That Can Be Entered as Commands in MARC

Command	Function
ABORT	Discontinues a job or task
ACCESS	Changes the accesscode password of the current job
ADD	Copies files to a disk, preserving resident versions
ARCHIVE	Makes a archive copy on tape of disk files
Assignment statement	Assigns a value to a variable
BEGIN	Indicates the start of a WFL subroutine
BIND	Runs the Binder utility to combine compiled code files
CASE	Selects one of several alternate WFL statements
CATALOG	Controls backup copies of disk and tape files
CHANGE	Changes the name of a disk file
COMPILE	Compiles, and optionally executes, a code file
COPY	Copies files to or from a disk or tape
CRUNCH	Makes available unused space at end of a disk file
DECK	Copies job control records to a disk file
DISPLAY	Displays a message at an ODT or user terminal
DO	Executes a WFL statement until a condition is met

continued

Table C-6. WFL Statements That Can Be Entered as Commands in MARC (cont.)

Command	Function
END	Indicates the end of a WFL subroutine
GO	Passes control of a WFL job to the statement at a label
IF	Executes a WFL statement if a condition is met
INITIALIZE	Reinitializes a task variable
INSTRUCTION	Supplies operator instructions for a job
LOCK	Closes a disk or tape file
LOG	Runs the LOGANALYZER utility to analyze the system log
MODIFY	Changes the attributes of a code file without recompiling the file
Null statement	Indicates that no action is requested
ON	Executes a WFL statement if a task terminates or restarts
OPEN	Opens a disk or tape file
PASSWORD	Changes the password for the usercode of a job
PB	Runs the BACKUP utility to output printer backup files
PRINT	Prints one or more files
PROCESS	Runs an asynchronous task
PTD	Runs the PTD utility to test peripheral devices
PURGE	Erases a file
RELEASE	Closes and releases a file
REMOVE	Deletes a disk file
RERUN	Restarts a program at a specified checkpoint
RETURN	Terminates a subroutine or ON statement
REWIND	Closes a disk or tape file, and sets a pointer to the beginning
RUN	Executes a code file
SECURITY	Changes the security of a disk file
START	Executes a WFL job from a disk file
STOP	Terminates a job or asynchronous subroutine
Subroutine invocation	Executes a previously declared subroutine
USER	Changes the usercode of a job
VOLUME	Controls disk or tape cataloging

continued

Commands Available in MARC

Table C-6. WFL Statements That Can Be Entered as Commands in MARC (cont.)

Command	Function
WAIT	Suspends a job until a condition is met
WHILE	Executes a WFL statement while a condition is met

Appendix D

Menu Selections for Common Tasks

Table D-1 lists the MARC menu selections for the most common system tasks. The tasks are arranged by category.

The table shows two sets of selections: one for users without SYSTEMUSER status and one for users with SYSTEMUSER status. If a task can be performed only by a user with SYSTEMUSER status, the *Not SYSTEMUSER* column is blank.

In each case the selection key is given first, followed by the menu name. For example, *SM on DC* refers to the SM selection on the DC menu.

If a security administrator has been defined for your system, tasks labeled *security administrator only* can be performed only by a user with security-administrator status. If no security administrator has been defined, such tasks can be performed by any user with SYSTEMUSER status.

See Section 7, "Controlling System Security," for more information about security privileges. For additional information about the tasks themselves, see the *A Series System Operations Guide*.

Table D-1. Menu Selections for Common Tasks

Category	Task	Selection (Not SYSTEMUSER)	Selection (SYSTEMUSER)
Data comm	Commands, sending to MCS		SM on DC
	Taking down data comm		QALL on DC
Database	Displaying stack entries		DBS on JD
	Sending commands to		SM on DC
Device	Acquiring		ACQ on PK, MT, and so on
	Clearing		CLEAR on PK, MT, and so on
	Directing output to		OU on JC
	Displaying status of		PER on PK, MT, and so on

continued

Menu Selections for Common Tasks

Table D-1. Menu Selections for Common Tasks (cont.)

Category	Task	Selection (Not SYSTEMUSER)	Selection (SYSTEMUSER)
	Displaying labels and paths		OL on PK, MT, and so on
	Readying		RY on PK, MT, and so on
	Releasing from the system		FREE on PK, MT, and so on
	Making inaccessible to the system		SV on PK, MT, and so on
Disks	Closing a disk unit		CLOSE on PK
	Consolidating space		SQUASH on PK
	Creating duplicate directories		DD on DSM
	Displaying available sectors		DU on FILE
	Purging		PG on PK
	Reconfiguring		MAKE on PK
	Scanning		SCAN on PK
Files	Archiving (backing up)	ARCF on ARCH	ARCF on ARCH
	Copying	COPY or OSCOPY on FILE	COPY or OSCOPY on FILE
	Creating	EDIT on FILE	EDIT on FILE
	Deleting	REMOVE on FILE	REMOVE on FILE
	Displaying directories	DIR or FDIR or PDIR on FILE	DIR or FDIR or PDIR on FILE
	Displaying information about	PDIR on FILE	PDIR on FILE
	Displaying libraries		ALL or LIBS on LIBS
	Rebuilding the file access structure table (FAST)		RB on DSM
Job queues	Creating		MQ on JQ
	Deleting		EQ on JQ
	Displaying attributes		QF on JQ

continued

Menu Selections for Common Tasks

Table D-1. Menu Selections for Common Tasks (cont.)

Category	Task	Selection (Not SYSTEMUSER)	Selection (SYSTEMUSER)
	Modifying		CQ on JQ
	Moving jobs within		MOVE on JQ
	Scheduling start times for jobs in		TIME on JQ
Jobs and tasks	Causing the exception event	HI on JDC	HI on JC
	Changing priority		PR on JC
	Determining status	Y on JDC	Y on JD
	Discontinuing	DS on JDC	DS on JC
	Displaying time accumulators	TI on JDC	TI on JD
	Forcing initiation		FS on JC
	Preventing initiation		HS on JC
	Passing text to	AX on JDC	AX on JC
	Restarting	OK on JDC	OK on JC
	Running tasks	RUN on MARC	RUN on MARC
	Starting WFL jobs	START on MARC	START on MARC
	Suspending		ST on JC
Memory	Dumping all memory		MEMDMP on DMPS
	Dumping task memory (program dump)		DMP or TDMP on DMPS
	Displaying memory usage		CU on MEM
Messages	Displaying session messages	MSG on JDC	MSG on JD
	Displaying system messages	SMSG on JDC	SMSG on JD
	Sending to other users	MAIL or SEND on MARC	MAIL or SEND on MARC
	Sending to database or MCS		SM on DC

continued

Menu Selections for Common Tasks

Table D-1. Menu Selections for Common Tasks (cont.)

Category	Task	Selection (Not SYSTEMUSER)	Selection (SYSTEMUSER)
Mix	Displaying active entries		A on JD
	Displaying completed entries	C on JDC	C on JD
	Displaying scheduled entries		S on JD
	Displaying structure	MX on JDC	MX on JD
	Setting the mix limit		SML on JQ
Print System	Deleting print requests	DELETE on PS	DELETE on PS
	Displaying configuration of a printer	SCON on DVCCTL	SCON on DVCCTL
	Displaying print request list	SHOW on PS	SHOW on PS
	Displaying printer attributes	DVCGRP on DVCCTL (privileged users only)	DVCGRP on DVCCTL
	Forcing print requests to print		FORCE on PS
	Handling exception print requests	MODIFY or DELETE on PS	MODIFY or DELETE on PS
	Modifying print requests	MODIFY on PS	MODIFY on PS
	Redirecting output to a printer	MODIFY on PS	MODIFY on PS
	Requeuing print requests	REQ on PS	REQ on PS
	Scheduling print requests	MODIFY on PS	MODIFY on PS
	Setting print server limits		SERV on PS
	Specifying location of backup files		SB on PS, then DLBACK on PS
	Stopping and restarting a printer	STOP on PS, then OK on PS	STOP on PS, then OK on PS

continued

Menu Selections for Common Tasks

Table D-1. Menu Selections for Common Tasks (cont.)

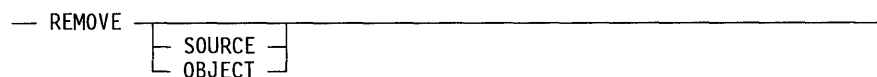
Category	Task	Selection (Not SYSTEMUSER)	Selection (SYSTEMUSER)
	Stopping and restarting the Print System		QUIT on PS, then RESTART on PS
Software installation	Installing keys files		IK on UTIL
	Installing system software		SI on UTIL (security administrator only)
System	Changing operating system version		CM on SYS
	Changing system date		DR on NDT
	Changing system time		TR on NDT
	Changing system time zone		TZC or TZP on NDT
	Creating alternate halt/load units		NHL on SYS
	Displaying current operating system version	WM on SYSINF	WM on SYS
	Displaying system configuration	SC on SYSINF	SC on CONFIG
	Displaying system utilization		BARS on UTIL or U on SYS
	Modifying USERDATAFILE		MU on UTIL (security administrator only)
	Powering off (A 1 to A 6 systems)		OFF on POWER
	Setting operating system options		SOP on SYS
Tapes	Assigning serial numbers		SN or SNL on MT
	Designating final reels		FR on JC
	Displaying directories		TDIR on MT
	Purging		PG or PGL on MT
	Rewinding		RW on MT

Appendix E

Understanding Railroad Diagrams

What Are Railroad Diagrams?

Railroad diagrams are diagrams that show you the rules for putting words and symbols together into commands and statements that the computer can understand. These diagrams consist of a series of paths that show the allowable structure, constants, and variables for a command or a statement. Paths show the order in which the command or statement is constructed. Paths are represented by horizontal and vertical lines. Many railroad diagrams have a number of different paths you can take to get to the end of the diagram. For example:



If you follow this railroad diagram from left to right, you will discover three acceptable commands. These commands are

- REMOVE
- REMOVE SOURCE
- REMOVE OBJECT

If all railroad diagrams were this simple, this explanation could end here. However, because the allowed ways of communicating with the computer can be complex, railroad diagrams sometimes must also be complex. Regardless of the level of complexity, all railroad diagrams are visual representations of commands and statements. Railroad diagrams are intended to

- Show the mandatory items.
- Show the user-selected items.
- Present the order in which the items must appear.
- Show the number of times an item can be repeated.
- Show the necessary punctuation.

To familiarize you with railroad diagrams, this explanation describes the elements of the diagrams and provides examples. Some of the actual railroad diagrams you will encounter might be more complex. However, all railroad diagrams, simple or complex, follow the same basic rules. They all consist of paths that represent the allowable structure, constants, and variables for commands and statements.

Understanding Railroad Diagrams

By following railroad diagrams, you can easily understand the correct syntax for commands and statements. Once you become proficient in the use of railroad notation, the diagrams serve as quick references to the commands and statements.

Constants and Variables

A constant is an item that cannot be altered. You must enter the constant as it appears in the diagram, either in full or as an allowable abbreviation. If a constant is partially underlined, you can abbreviate the constant by entering only the underlined letters. In addition to the underlined letters, any of the remaining letters can be entered. If no part of the constant is underlined, the constant cannot be abbreviated. Constants can be recognized by the fact that they are never enclosed in angle brackets (< >) and are in uppercase letters.

A variable is an item that represents data. You can replace the variable with data that meets the requirements of the particular command or statement. When replacing a variable with data, you must follow the rules defined for the particular command or statement. Variables appear in railroad diagrams enclosed in angle brackets.

In the following example, BEGIN and END are constants while <statement list> is a variable. The constant BEGIN can be abbreviated since it is partially underlined. Valid abbreviations for BEGIN are BE, BEG, and BEGI.

— BEGIN —<statement list>— END —————|

Constraints

Constraints are used in a railroad diagram to control progression through the diagram. Constraints consist of symbols and unique railroad diagram line paths. They include

- Vertical bars
- Percent signs
- Right arrows
- Required items
- User-selected items
- Loops
- Bridges

A description of each item follows.

Vertical Bar

The vertical bar symbol (|) represents the end of a railroad diagram and indicates the command or statement can be followed by another command or statement.

— SECONDWORD — (—<arithmetic expression>—) —————|

Percent Sign

The percent sign (%) represents the end of a railroad diagram and indicates the command or statement must be on a line by itself.

— STOP ————— %

Right Arrow

The right arrow symbol (>) is used when the railroad diagram is too long to fit on one line and must continue on the next. A right arrow appears at the end of the first line, and another right arrow appears at the beginning of the next line.

— SCALERIGHT — (—<arithmetic expression>— , —————>
 >—<arithmetic expression>—) —————|

Required Items

A required item can be either a constant, a variable, or punctuation. A required item appears as a single entry, by itself or with other items, on a horizontal line. Required items can also exist on horizontal lines within alternate paths or nested (lower-level) diagrams. If the path you are following contains a required item, you must enter the item in the command or statement; the required item cannot be omitted.

In the following example, the word EVENT is a required constant and <identifier> is a required variable:

— EVENT —<identifier>—————|

User-Selected Items

User-selected items appear one below the other in a vertical list. You can choose any one of the items from the list. If the list also contains an empty path (solid line), none of the choices are required. A user-selected item can be either a constant, a variable, or punctuation. In the following railroad diagram, either the plus sign (+) or the minus sign (-) can be entered before the required variable <arithmetic expression>, or the symbols can be disregarded because the diagram also contains an empty path.

	+	
	-	

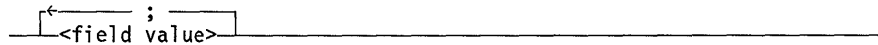
 <arithmetic expression>—————|

Understanding Railroad Diagrams

Loop

A loop represents an item or group of items that you can repeat. A loop can span all or part of a railroad diagram. It always consists of at least two horizontal lines, one below the other, connected on both sides by vertical lines. The top line is a right-to-left path that contains information about repeating the loop.

Some loops include a return character. A return character is a character – often a comma (,) or semicolon (;) – required before each repetition of a loop. If there is no return character, the items must be separated by one or more blank spaces.

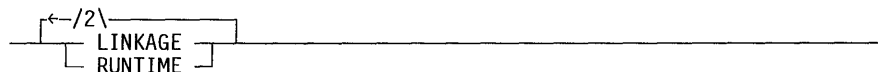
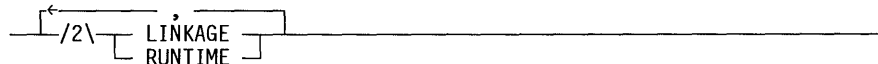


Bridge

Sometimes a loop also includes a bridge, which is used to show the maximum number of times the loop can be repeated. The bridge can precede the contents of the loop, or it can precede the return character (if any) on the upper line of the loop.

The bridge determines the number of times you can cross that point in the diagram. The bridge is an integer enclosed in sloping lines (/ \). Not all loops have bridges. Those that do not can be repeated any number of times until all valid entries have been used.

In the first bridge example, you can enter LINKAGE or RUNTIME no more than two times. In the second bridge example, you can enter LINKAGE or RUNTIME no more than three times.



In some bridges an asterisk (*) follows the number. The asterisk means that you must cross that point in the diagram at least once. The maximum number of times that you can cross that point is indicated by the number in the bridge.



In the previous bridge example, you must enter LINKAGE at least once but no more than twice, and you can enter RUNTIME any number of times.

The following figure shows the types of constraints used in railroad diagrams.



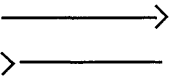
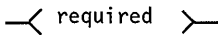
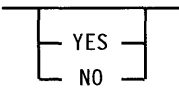
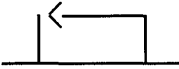
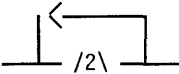
SYMBOL/PATH	EXPLANATION
	Vertical bar. Indicates that the command or statement can be followed by another command or statement.
	Percent sign. Indicates that the command or statement must be on a line by itself.
	Right arrow. Indicates that the diagram occupies more than one line.
	Required items. Indicates the constants, variables, and punctuation that must be entered in a command or statement.
	User-selected items. Indicates the items that appear one below the other in a vertical list. You select which item or items to include.
	A loop. Indicates an item or group of items that can be repeated.
	A bridge. Indicates the maximum number of times a loop can be repeated.

Figure E-1. Railroad Constraints

Following the Paths of a Railroad Diagram

The paths of a railroad diagram lead you through the command or statement from beginning to end. Some railroad diagrams have only one path, while others have several alternate paths. The following railroad diagram indicates there is only one path that requires the constant LINKAGE and the variable <linkage mnemonic>:

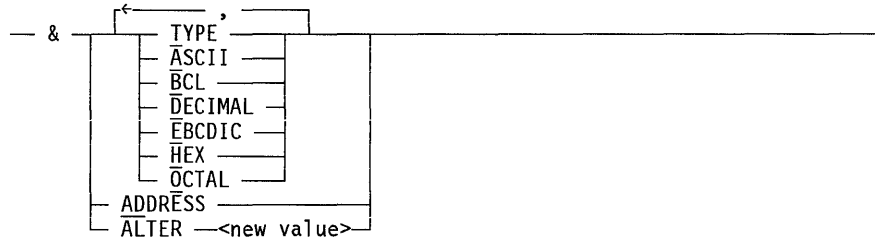
— LINKAGE —<linkage mnemonic>—————|

Alternate paths provide choices in the construction of commands and statements. Alternate paths are provided by loops, user-selected items, or a combination of both. More complex railroad diagrams can consist of many alternate paths, or nested (lower-level) diagrams, that show a further level of detail.

For example, the following railroad diagram consists of a top path and two alternate paths. The top path includes an ampersand (&) and the constants (that are

Understanding Railroad Diagrams

user-selected items) in the vertical list. These constants are within a loop that can be repeated any number of times until all options have been selected. The first alternate path requires the ampersand and the required constant ADDRESS. The second alternate path requires the ampersand followed by the required constant ALTER and the required variable <new value>.



Railroad Diagram Examples with Sample Input

The following examples show five railroad diagrams and possible command and statement constructions based on the paths of these diagrams.

Example 1

<lock statement>



Sample Input

LOCK (FILE4)

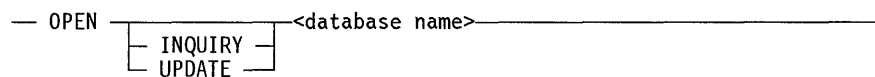
Explanation

LOCK is a constant and cannot be altered. Because no part of the word is underlined, the entire word must be entered.

The parentheses are required punctuation, and FILE4 is a sample file identifier.

Example 2

<open statement>



Sample Input

OPEN DATABASE1

Explanation

The constant OPEN is followed by the variable DATABASE1, which is a database name.

The railroad diagram shows two user-selected items, INQUIRY and UPDATE. However, because there is an empty path (solid line), these entries are not required.

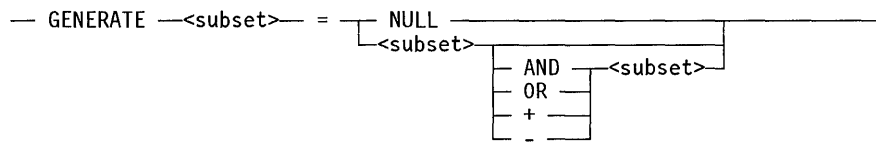
continued

continued

Sample Input	Explanation
OPEN INQUIRY DATABASE1	The constant OPEN is followed by the user-selected constant INQUIRY and the variable DATABASE1.
OPEN UPDATE DATABASE1	The constant OPEN is followed by the user-selected constant UPDATE and the variable DATABASE1.

Example 3

<generate statement>

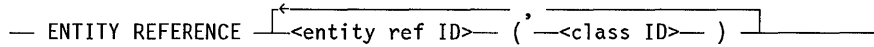


Sample Input	Explanation
GENERATE Z = NULL	The GENERATE constant is followed by the variable Z, an equal sign (=), and the user-selected constant NULL.
GENERATE Z = X	The GENERATE constant is followed by the variable Z, an equal sign, and the user-selected variable X.
GENERATE Z = X AND B	The GENERATE constant is followed by the variable Z, an equal sign, the user-selected variable X, the AND command (from the list of user-selected items in the nested path), and a third variable, B.
GENERATE Z = X + B	The GENERATE constant is followed by the variable Z, an equal sign, the user-selected variable X, the plus sign (from the list of user-selected items in the nested path), and a third variable, B.

Understanding Railroad Diagrams

Example 4

<entity reference declaration>



Sample Input

ENTITY REFERENCE ADVISOR1
(INSTRUCTOR)

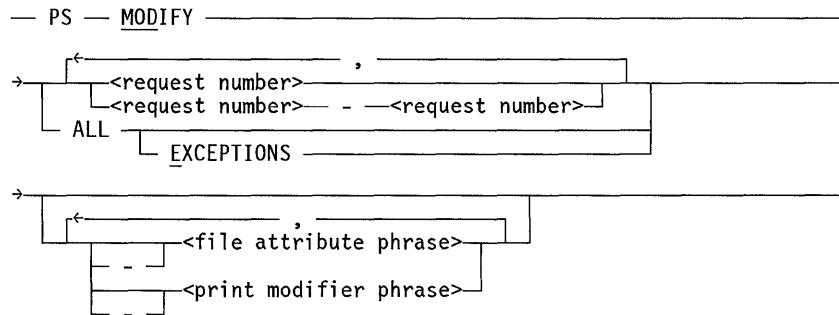
Explanation

The required item ENTITY REFERENCE is followed by the variable ADVISOR1 and the variable INSTRUCTOR. The parentheses are required.

ENTITY REFERENCE ADVISOR1
(INSTRUCTOR), ADVISOR2
(ASST_INSTRUCTOR)

Because the diagram contains a loop, the pair of variables can be repeated any number of times.

Example 5



Sample Input

PS MODIFY 11159

Explanation

The constants PS and MODIFY are followed by the variable 11159, which is a request number.

PS MODIFY
11159,11160,11163

Because the diagram contains a loop, the variable 11159 can be followed by a comma, the variable 11160, another comma, and the final variable 11163.

PS MOD 11159-11161
DESTINATION = "LP7"

The constants PS and MODIFY are followed by the user-selected variables 11159-11161, which are request numbers, and the user-selected variable DESTINATION = "LP7", which is a file attribute phrase. Note that the constant MODIFY has been abbreviated to its minimum allowable form.

PS MOD ALL EXCEPTIONS

The constants PS and MODIFY are followed by the user-selected constants ALL and EXCEPTIONS.

Glossary

A

accesscode

An identification code, subordinate to a usercode, which sometimes has an associated password of its own. An accesscode can be specified in the USERDATAFILE as required along with a usercode/password combination. An accesscode further establishes a user's identity, controls security, and restricts access to disk files.

Action field

A field that appears on a form or screen and begins with the prompt *Action:*. The user enters the desired action for execution between the indicators to the right of this prompt.

action word

In Unisys menu-based software, a word that can be entered in the Action field. Action words can be only those words listed below the Action field.

actual segment descriptor (ASD)

A pointer to the location of a data or code item in memory or on a disk.

ADDS

See Advanced Data Dictionary System.

Advanced Data Dictionary System (ADDS)

A software product that allows for the centralized definition, storage, and retrieval of data descriptions.

agenda

In the Communications Management System (COMS), an entity used for message routing that consists of a processing-item list and a destination. An agenda can be applied to messages that are received or sent by application programs.

ALGOL

Algorithmic language. A structured, high-level programming language that provides the basis for the stack architecture of the Unisys A Series systems. ALGOL was the first block-structured language developed in the 1960s and served as a basis for such languages as Pascal and Ada. It is still used extensively on A Series systems, primarily for systems programming.

ASD

See actual segment descriptor.

B

beginning of task (BOT)

The start of processing of a task.

BNA

The network architecture used on A Series, B 1000, and V Series systems as well as CP9500 and CP 2000 communications processors to connect multiple, independent, compatible computer systems into a network for distributed processing and resource sharing.

book cache

An internal structure, maintained by the HELPSUPPORT library, that stores recently accessed help books.

BOT

See beginning of task.

C

CANDE

See Command and Edit.

ccsversion

The instructions for the application of a coded character set, including information about the way in which data is presented, collated, and cased.

CD

See communication description, compact disk.

character set

(1) (ANDIPS) A finite set of different characters upon which agreement has been reached and that is considered complete for some purpose. For example, in COBOL programming, the character set contains 52 specific characters. (2) The letters, numbers, symbols, and special characters used to represent data in a particular locality or line of business.

chargecode

An account number to which the cost of a computer session is billed. System administrators can determine how chargecodes are used on their particular systems.

Choice field

A field on a menu in which a user enters a selection key to make a menu selection. The Choice field begins with the prompt *Choice:* and usually appears near the bottom of the screen or form. The selection key is entered between the indicators to the right of this prompt.

coded character set

A character set in which each character is assigned a code value.

Command and Edit (CANDE)

A time-sharing message control system (MCS) that enables a user to create and edit files, and to develop, test, and execute programs, interactively.

command mode

A mode of operation that enables the user to enter commands directly and bypass the normal sequence of screens or menus. *Contrast with* menu mode.

communication description (CD)

An obsolete term for the COMS_HEADER parameter that is passed with the message data received and sent by application programs that interface with a message control system (MCS).

Communications Management System (COMS)

A general message control system (MCS) that controls online environments on A Series systems. COMS can support the processing of multiprogram transactions, single-station remote files, and multistation remote files. *See also* COMS (Full-Featured) and COMS (Kernel).

COMS

See Communications Management System.

COMS (Full-Featured)

A version of the Communications Management System (COMS) message control system (MCS) that provides full configuration capabilities through the COMS Utility. The COMS Utility enables the user to define and customize a COMS transaction processing environment, which provides additional features like transaction-based routing and database recovery. In addition, the user can track COMS statistics and use GEMCOS migration aids.

COMS header

In Communications Management System (COMS), the part of the communication structure that contains routing and descriptive information about the message. There is an input header for input messages and an output header for output messages.

COMS_HEADER parameter

A parameter passed with the message data received and sent by application programs that interface with a message control system (MCS). The parameter provides routing information about the message data and allows use of other functions, depending on the MCS.

COMS (Kernel)

The transitional, temporary version of the Communications Management System (COMS) message control system (MCS). COMS creates a predefined configuration file that enables the user to use the window feature with the following three windows: a Menu-Assisted Resource Control (MARC) window with eight dialogues, a Command and Edit (CANDE) window with two dialogues, and a Generalized Message Control System (GEMCOS) window with one dialogue. Additionally, the user can communicate with remote-file programs.

Glossary

COMS Utility

The Communications Management System (COMS) program that defines and maintains the specifications stored in the COMS configuration file.

control station

In the Command and Edit (CANDE) message control system (MCS), a station that allows CANDE network control commands to be entered.

control-capable station

In the Communications Management System (COMS), a station with no restrictions on the use of COMS commands.

convention definition

A description used to build the format and content of a convention.

convention set

A collection of convention definitions that are always accessed as a single unit.

conventions

(1)The standards and accepted procedures that are developed for a particular system, language, or program. (2)The agreed-upon formats for presenting date, time, numeric, and monetary information as well as the default number of lines per page, and the default number of characters per line.

conversation area

In the Communications Management System (COMS), the user data space in the header of a message. The conversation area is user defined and can contain information passed by a program or processing item. When used with a direct-window interface, this area contains the telephone number to be dialed.

CTRL (control) key

A special function key on some keyboards that generates control sequences for transmission to the system.

cursor

The marker on the terminal screen that indicates where the next character entered appears.

D

data comm mode

A mode of operation at an operator display terminal (ODT) in which interaction is controlled by the Communications Management System (COMS). The operator can use this mode to access COMS windows such as the Menu-Assisted Resource Control (MARC) interface or the Command and Edit (CANDE) message control system (MCS).

data comm station

In BNA, a station connected to a network support processor (NSP) or data communications data link processor (DCDLP) that is being used as a terminal.

dialogue

(1) In interprocess communication, a single instance of a two-way communication between two processes. A port subfile supports one dialogue. (2) *See* window dialogue.

directive command

In the Menu-Assisted Resource Control (MARC) interface, a command that performs a user-defined, user-created, and user-named procedure, and that is entered in the same way as any other MARC command.

DLS number

A construct made up of three numbers, each separated by a colon (:). The first number is the relative network support processor (NSP) number, which was previously the data communications processor (DCP) number. The second number is the line number. The third number is the relative station number within the line.

E

EBCDIC

Extended Binary Coded Decimal Interchange Code. An 8-bit code representing 256 graphic and control characters that are the native character set of most mainframe systems.

editor

A Network Definition Language II (NDLII) program module that defines an input process and an output process. These processes implement application-dependent editing on the text portions of input and output messages according to the requirements of specific terminal types.

embedded keyword

A keyword that appears within help text. An embedded keyword is linked to other help text that supplies the user with information about the embedded keyword.

end of task (EOT)

The termination of processing of a task.

end of transmission (EOT)

A control code that tells the receiver that all user data (text) has been sent.

EOT

See end of task, end of transmission.

ERGO

See Extended Retrieval with Graphic Output.

F

field

(1) An area on a screen or form in which data is displayed or entered. The delimiters of the field can be visible or invisible to the terminal operator. (2) A consecutive group of bits within a word or a component of a record that represents a logical piece of data.

Glossary

form

A special screen containing prompts requesting information and empty form fields in which the requested information can be entered.

form field

A specific area on a screen or form that either contains text or is intended for the entry of text. A form field is bordered by solid triangles, square brackets ([]), or another delineating character, depending on the terminal type.

forms mode

A method of operating at a terminal in which the placement of input fields is identified by the system.

forms-mode capability

A feature offered by those terminals that can be configured to recognize the placement of input fields.

full-screen transmission

A terminal configuration in which the terminal transmits the data in all the unprotected fields on the screen when the transmit key is pressed, regardless of the cursor position.

G

GEMCOS

See Generalized Message Control System.

Generalized Message Control System (GEMCOS)

A message control system (MCS) developed for online systems. GEMCOS is transaction oriented.

guard file

A disk file created by the GUARDFILE utility program that describes the access rights of various users and programs to a program, data file, or database.

H

header

(1) A data structure that contains information about a disk file, such as the physical location of the file on the disk and various file attributes. A header is also referred to as a disk file header. (2) A sequence of characters preceding the text of a message, containing routing or other communications-related information.

help

A method of operation in Menu-Assisted Resource Control (MARC) and similar screen-based products in which help text is displayed by pressing the SPCFY (specify) key on a selection or in a field, or by entering HELP or TEACH along with a keyword.

help book

A file of help text that is generated by the Help Utility. The help text is intended to provide online information for users. The help book is accessed at run time when a user presses the SPCFY (specify) key or enters HELP or TEACH along with a keyword.

help book source file

A file containing formatted text and embedded help utility commands that can be processed by a help utility to produce a help book.

HELP command

An instruction that requests help information associated with the specified selection key or keyword. TEACH is a synonym for HELP in this command. HELP commands can be entered in the Action field or the Choice field of any Menu-Assisted Resource Control (MARC) menu or similar screen. All the help information that can be accessed by pressing the SPCFY (specify) key twice while the cursor is on a MARC menu or similar screen can also be accessed by using the HELP command.

help information

(1) A detailed explanation of an item on a screen. (2) In the Interactive Datacomm Configurator (IDC), a detailed explanation of an item on a screen accessible only when IDC is in interactive mode.

help library

A library that contains help text.

help message

In the Screen Design Facility Plus (SDF Plus), a message that provides an explanation of a particular form, field, embedded keyword, or system message and that can be accessed online by the user. *Synonym for help text.*

help reference

In the Screen Design Facility Plus (SDF Plus), a structure that identifies a help message and the help library in which the message is stored. The help reference name links a help message to a user interface system (UIS) component.

help screen

In the Data Transfer System (DTS), a screen made available for most functions by pressing the HELP key. Help screens give the user the necessary information to complete a particular task without interrupting the progress of that task.

help text

Information about forms, screens, windows, fields, embedded keywords, and system messages available to a user online.

Help Utility

One element of a system designed to display online information for the computer user. The function of the Help Utility is to create a book that contains help text and keywords.

Glossary

hexadecimal (hex)

Pertaining to the base 16 numbering system. Decimal digits 0 through 9 are represented by the characters 0 through 9. Decimal digits 10 through 15 are represented by the characters A through F.

hexadecimal character

One of a set of characters that includes 0 through 9 and A through F used to represent quantities in the base 16 numbering system. The characters A through F represent the decimal values 10 through 15, respectively.

I

IDC

See Interactive Datacomm Configurator.

IMG

See Interactive Menugraph Generator.

input field

A form field that can accept text as input to the computer.

Interactive Datacomm Configurator (IDC)

A Unisys interactive, menu-driven utility that enables the user to create, interrogate, and modify data communications network configurations.

interactive definition module (IDM)

The online, interactive interface of the Advanced Data Dictionary System (ADDS).

Interactive Menugraph Generator (IMG)

A software tool for the design and modification of Menu-Assisted Resource Control (MARC), the Interactive Datacomm Configurator (IDC), and IMG screen menus and forms.

internal form

In the Interactive Menugraph Generator (IMG), a form that has no command linked with it. Instead, an internal form is used for noncommand purposes, such as supplying a template for the help text screen.

J

job

An independent process. The job of a particular task is the independent process that is the eldest ancestor of that task.

K

keyword

A word or group of words that supplies the link between help text and those screen items for which the information is intended.

L**library**

A collection of one or more named routines or library objects that are stored in a file and can be accessed by other programs.

localization

The act of adapting a system or product so that it complies with the standards and functional requirements of a particular locality or line of business.

logical station number (LSN)

A unique number assigned to each station in a network and each pseudostation allocated by a message control system (MCS). Each station has an LSN assigned according to the order in which the stations are defined.

long help text

The detailed, online explanations of a form, screen, field, or window. Long help text is invoked in different ways by different products and appears in a window or on a separate page that temporarily replaces the screen from which help was invoked.

LSN

See logical station number.

M**MARC**

See Menu-Assisted Resource Control.

master control program (MCP)

The central program of the A Series operating system. The term applies to any master control program that Unisys may release for A Series systems.

MCP

See master control program.

MCS

See message control system.

menu

A special group of fields that presents the user with a set of options from which to choose.

menu mode

The mode of operation in which the user selects options on a menu. *Contrast with* command mode.

menu selection

The option the user chooses on a menu.

Glossary

Menu-Assisted Resource Control (MARC)

A menu-driven interface to A Series systems that also enables direct entry of commands.

menugraph

In the Interactive Menugraph Generator (IMG), a file containing definitions of a linked system of menu-based screens arranged in a hierarchy.

message control system (MCS)

A program that controls the flow of messages between terminals, application programs, and the operating system. MCS functions can include message routing, access control, audit and recovery, system management, and message formatting.

Message Handling System (MHS)

A distributed systems service (DSS) that implements the Consultative Committee on International Telegraphy and Telephony (CCITT) X.400 standards for store-and-forward message handling systems.

Message Translation Utility (MSGTRANS)

An A Series product used to localize messages from software products in a MultiLingual System (MLS) environment. MSGTRANS translates compiled program messages from any natural language to one or more other natural languages. MSGTRANS can be used to edit or translate messages, access messages from an external file, print messages, save messages, and bind them to the original software product.

MHS

See Message Handling System.

MLS

See MultiLingual System.

monitor station

In the Communications Management System (COMS), a station that has been defined to display COMS activities specified as monitor options.

MSGTRANS

See Message Translation Utility.

multilingual system (MLS)

A system for developing and accessing output messages, online help text, and menu screens in different natural languages, such as English, French, and Spanish.

N

NDLII

See Network Definition Language II.

NETEX

A host-to-host communications package that enables application programs in different hosts to communicate with each other, regardless of specific network configuration.

Through the NETEX software, applications such as file transfer and transaction processing can be implemented across hosts.

Network Definition Language II (NDLII)

The Unisys language used to describe the physical, logical, and functional characteristics of the data communications subsystem to network support processors (NSPs), line support processors (LSPs), and data communications data link processors (DCDLPs).

O

ODT

See operator display terminal.

operator display terminal (ODT)

A terminal or other device that is connected to the system in such a way that it can communicate directly with the operating system. The ODT allows operations personnel to accomplish system operations functions through either of two operating modes: system command mode or data comm mode.

Open Systems Interconnection (OSI)

A set of data communications standards defined by the International Organization for Standardization (ISO) that provide for communications between different types of computer systems. The application services defined under OSI include File Transfer, Access, and Management (FTAM) and the Message Handling System (MHS).

OSI

See Open Systems Interconnection.

P

parent screen

The screen that most logically precedes the current screen when a user traverses the screens of a menugraph. A parent screen is also referred to as the parent.

password

A character string associated with a usercode or accesscode in the USERDATAFILE, and used to identify legitimate users of the system. When logging on to a message control system (MCS), a user must supply a usercode and a password.

PC

See personal computer.

peripheral test driver (PTD)

A module of the master control program (MCP) that executes maintenance tests for peripheral devices.

Glossary

personal computer (PC)

A microcomputer that is compatible with the Unisys Personal Workstation ², the IBM PC®, the PC/XT™, the Personal Computer AT®, or the IBM Personal System/2® line of computers.

PIB

See process information block.

Print System (PrintS)

A Unisys software product used to control when, where, and how printer backup files are printed on A Series systems.

process information block (PIB)

A memory structure that is associated with each process stack, and which stores the values of the task attributes of that process.

processing item

A procedure, contained in a processing-item library, used for processing a message.

prompt

The screen text that asks the user to enter a response in an associated I/O field. Prompts aid the user in using screens correctly.

PTD Utility

See peripheral test driver.

R

railroad diagram

A graphic representation of the syntax of a command or statement.

remote file

A file with the KIND attribute specified as REMOTE. A remote file enables object programs to communicate interactively with a terminal.

Remote Print System (ReprintS)

A Unisys software system that controls the routing and printing of backup files at remote (data comm) destinations and on BNA networks.

S

screen

An image that appears on the display area of a terminal or workstation and performs at least one of the following functions: prompting the user to enter data, displaying information, or presenting options from which to choose.

PC/XT is a trademark of International Business Machines Corporation.

Personal Computer AT is a registered trademark of International Business Machines Corporation.

IBM Personal System/2 is a registered trademark of International Business Machines Corporation.

screen cache

In Menu-Assisted Resource Control (MARC), an internal structure that stores the last 15 screens that were displayed. For help text screens, the last 10 screens are stored.

Screen Design Facility (SDF)

The InterPro product used for creating forms for online, transaction-based application systems.

Screen Design Facility Plus (SDF Plus)

A Unisys product used for creating user interface systems (UIs) for online, transaction-based application systems.

screen name

A unique identifier for a particular screen.

scroll

To move forward and backward within a list, within help text, or within other items that can be displayed.

SCT

See system control terminal.

SDF

See Screen Design Facility.

SDF Plus

See Screen Design Facility Plus.

secure-video mode

On Unisys TD-like terminals, a method of preventing a password from being visible on the screen as the user types it in.

selection description

A brief description of the function of a menu selection or the result of making a menu selection. In a column of menu selections, the selection description appears on the right and the associated selection key appears on the left.

selection key

A string of six or fewer characters that indicates a particular menu selection. In a column of menu selections, the selection keys appear on the left and a short description of each selection appears on the right.

short help text

A brief, online explanation of a form, screen, window, or field.

SI

See Simple Installation.

sibling

A task that has the same parent as another task.

Glossary

Simple Installation (SI)

A program that simplifies the installation of Unisys system software through either batch commands or menu screens.

single-line mode

The method of transmission in which the maximum message size is the length of one line.

source file

In the Help Utility, the file containing embedded commands and help text. A source file is designed to be processed by the Help Utility to create a book.

SPCFY (specify) key

A function key on some keyboards that transmits the cursor position to the system.

station

The outer end of a communication line. A station can correspond to a single terminal connected on a single line, or several stations can be connected on a line.

super-user-capable station

In the Communications Management System (COMS), a station at which any user can log on to MARC by specifying an asterisk (*) in the usercode field of the LOGON screen, instead of supplying a usercode and password. The asterisk (*) usercode has SYSTEMUSER status, but is not privileged.

super user

A usercode security category like privileged or SYSTEMUSER, except that super-user status is not associated with an individual usercode. Instead, the status is associated with a particular station. A user has super-user status only when he or she accesses the system from a super-user-capable station, and he or she enters an asterisk (*) in the usercode field during the log-on sequence.

system command

Any of a set of commands used to communicate with the operating system. System commands can be entered at an operator display terminal (ODT), in a Menu-Assisted Resource Control (MARC) session, or by way of the DCKEYIN function in a privileged Data Communications ALGOL (DCALGOL) program.

system command mode

A mode of operation at an operator display terminal (ODT) in which operations personnel can enter system commands and view automatic display mode (ADM) displays.

system control terminal (SCT)

(1) A terminal or other device that is connected to the system in such a way that it can communicate directly with the maintenance processor. An SCT can operate in maintenance mode or in operator display terminal (ODT) mode. On some systems, the SCT also provides a remote support mode. (2) A terminal used to enter information. An SCT can be used three ways: as an operator display terminal (ODT) to interface with the operating system, as a maintenance display terminal (MDT) to interface with the maintenance subsystem, or as a remote display terminal (RDT) to interface with

remote support. The windows providing these uses are available once the automatic initialization sequence has finished.

system log file

A file that contains a record of events for a particular system.

T**TADS**

See Test and Debug System.

tanking

(1) In the transaction processing system (TPS), the operation in which the transaction processor (TP) library stores transactions in a tank journal and does not process them against the database. A tank journal can be any transaction journal except the TRHISTORY journal. (2) The practice of temporarily storing output messages in a disk file because the destination station is unavailable. The operating system and the Communications Management System (COMS) both perform tanking. (3) The practice of temporarily storing messages from a Data Communications ALGOL (DCALGOL) queue in a disk file until the receiving process is ready to read the messages.

task

(1) A dependent process. (2) Any process, whether dependent or independent. *See also* process.

tasking

(1) The act of initiating, monitoring, or controlling processes. The processes can be either jobs or tasks. Operators and users can enter tasking commands from an operator display terminal (ODT), a Command and Edit (CANDE) session, or a Menu-Assisted Resource Control (MARC) session. Programs can initiate processes with such statements as CALL, PROCESS, or RUN. Programs can monitor and control processes by reading and assigning the values of various task attributes. (2) In the Menu-Assisted Resource Control (MARC) interface, the act of monitoring or interacting with a process through the Task Status screen, the Task View screen, or the task window. MARC provides commands that enable the user to enter and leave tasking easily and *move about* within tasking.

TD-like terminal

A terminal that has the characteristics of a Unisys TD 830 terminal. For example, the MT 983 and ET 1100 are TD-like terminals.

Teletype-like terminal

A terminal that has the line-at-a-time transmission characteristics of a Teletype machine.

terminal

An I/O device designed to receive or send source data in a network.

Glossary

Test and Debug System (TADS)

A Unisys interactive tool for testing and debugging programs and libraries. TADS enables the programmer to monitor and control the execution of the software under testing and examine the data at any given point during program execution.

typeahead

In Menu-Assisted Resource Control (MARC), a capability that allows a sequence of entries in the Action or Choice fields. The entries can be used to bypass a series of screens.

U

UCF

See User Communication Form.

User Communication Form (UCF)

A form used by Unisys customers to report problems and express comments about Unisys products to support organizations.

usercode

An identification code used to establish user identity and control security, and to provide for segregation of files. Usercodes can be applied to every task, job, session, and file on the system. A valid usercode is identified by an entry in the USERDATAFILE.

W

WFL

See Work Flow Language.

WFL job

(1) A Work Flow Language (WFL) program, or the execution of such a program. (2) A collection of Work Flow Language (WFL) statements that enable the user to run programs or tasks.

window

(1) In the Communications Management System (COMS) architecture, the concept that enables a number of program environments to be operated independently and simultaneously at one station. One of the program environments can be viewed while the others continue to operate. (2) In the Editor, a conceptual aperture through which any screen-sized group of lines in a work file can be viewed. The window is said to move forward in the file if lines of higher sequence numbers are being displayed, and backward if lines of lower sequence numbers are being displayed. (3) A portion of a screen that has been allocated to display the contents of a specified area of memory.

Work Flow Language (WFL)

A Unisys language used for constructing jobs that compile or run WFL includes variables, expressions, and flow-of-control statements that offer the programmer

X

XMIT (transmit) key

The key on the terminal keyboard that, when pressed, transmits the entered data to the computer system. Synonym for XMT (transmit) key.

Bibliography

- A Series A 1 – A 6 Systems Software Installation Guide* (8600 1468). Unisys Corporation.
- A Series ALGOL Programming Reference Manual, Volume 1: Basic Implementation* (8600 0098). Unisys Corporation.
- A Series ALGOL Test and Debug System (TADS) Programming Guide* (1169539). Unisys Corporation.
- A Series BNA Version 1 Operations Guide* (8600 0783). Unisys Corporation.
- A Series CANDE Operations Reference Manual* (8600 1500). Unisys Corporation.
- A Series COBOL ANSI-74 Test and Debug System (TADS) Programming Guide* (1169901). Unisys Corporation.
- A Series Communications Management System (COMS) Configuration Guide* (8600 0312). Unisys Corporation.
- A Series Communications Management System (COMS) Operations Guide* (8600 0833). Unisys Corporation.
- A Series Communications Management System (COMS) Programming Guide* (8600 0650). Unisys Corporation.
- A Series Data Management Functional Overview* (8600 0239). Unisys Corporation.
- A Series DMSII Utilities Operations Guide* (8600 0759). Unisys Corporation.
- A Series Documentation Library Overview* (8600 0361). Unisys Corporation.
- A Series Editor Operations Guide* (8600 0551). Unisys Corporation.
- A Series Extended Retrieval with Graphic Output (ERGO) Operations Guide* (8600 0205). Unisys Corporation.
- A Series File Attributes Programming Reference Manual* (8600 0064). Unisys Corporation.
- A Series FORTRAN77 Test and Debug System (TADS) Programming Guide* (1222667). Unisys Corporation.
- A Series Help Utility Operations Guide* (8600 0510). Unisys Corporation.
- A Series InfoExec Advanced Data Dictionary (ADDS) Operations Guide* (8600 0197). Unisys Corporation.

Bibliography

- A Series Interactive Datacomm Configurator (IDC) Operations Guide* (1169810). Unisys Corporation.
- A Series Interactive Menugraph Generator (IMG) Operations Guide* (8600 0411). Unisys Corporation.
- A Series Message Translation Utility (MSGTRANS) Operations Guide* (8600 0106). Unisys Corporation.
- A Series Micro A System Installation Guide* (5044100). Unisys Corporation.
- A Series Model S System Software Installation Guide* (5044423). Unisys Corporation.
- A Series MultiLingual System (MLS) Administration, Operations, and Programming Guide* (8600 0288). Unisys Corporation.
- A Series Network Definition Language II (NDLII) Programming Reference Manual* (1169604). Unisys Corporation.
- A Series PCMARC Installation and Operations Guide* (8600 0817). Unisys Corporation.
- A Series Print System (PrintS/ReprintS) Administration, Operations, and Programming Guide* (8600 1039). Unisys Corporation.
- A Series Printing Utilities Operations Guide* (8600 0692). Unisys Corporation.
- A Series Screen Design Facility (SDF) Operations and Programming Guide* (1185295). Unisys Corporation.
- A Series Screen Design Facility Plus (SDF Plus) Installation and Operations Guide* (8600 0262). Unisys Corporation.
- A Series Security Administration Guide* (8600 0973). Unisys Corporation.
- A Series Security Features Operations and Programming Guide* (8600 0528). Unisys Corporation.
- A Series Software Release Installation Guide* (8600 0981). Unisys Corporation.
- A Series System Administration Guide* (8600 0437). Unisys Corporation.
- A Series System Commands Operations Reference Manual* (8600 0395). Unisys Corporation.
- A Series System Configuration Guide* (8600 0445). Unisys Corporation.
- A Series System Messages Support Reference Manual* (8600 0429). Unisys Corporation.
- A Series System Operations Guide* (8600 0387). Unisys Corporation.

A Series System Software Support Reference Manual (8600 0478). Unisys Corporation.

A Series System Software Utilities Operations Reference Manual (8600 0460). Unisys Corporation.

A Series Systems Functional Overview (8600 0353). Unisys Corporation.

A Series Task Attributes Programming Reference Manual (8600 0502). Unisys Corporation.

A Series Task Management Programming Guide (8600 0494). Unisys Corporation.

A Series Work Flow Language (WFL) Programming Reference Manual (8600 1047). Unisys Corporation.

Index

A

A Series

- billing support facility, 1-3, 6-42, B-7
 - computer system, 1-1
 - documents, viii, ix, x, xi
 - online, 1-5, 3-7
 - Editor, 1-6
 - Help Utility, 1-6
 - Mail System, 4-6
 - Master Glossary, online, 1-5, 3-4, 3-7
 - message suppression feature, 6-17
 - security categories, 7-1
 - terms, online, 1-5, 3-4, 3-7
 - user, 1-1
- accesscode, 2-3
- password changes, 7-6
 - screen, bypassing, 2-3
- Action field, 2-6
- choices, (See actions)
 - input overriding other fields, B-6
 - running a task from, 4-10
 - starting a WFL job from, 4-4
 - typeahead, 2-8
- action words
- localizing, 9-12
 - tasking, 4-13
- actions, 2-13
- COL, 2-16
 - COmnd, 2-14, 6-1
 - FIrst, 2-15
 - GO, 2-14
 - HOme, 2-13
 - to get the default menugraph, 8-11
 - KEys, 2-14
 - LAst, 2-16
 - PArent, 2-14
 - PREv, 2-13
 - after using the LANGUAGE command, 6-13
 - REturn, 2-15
 - scrolling backward, 2-15
 - scrolling forward, 2-15

- SEQ, 2-16
 - + (plus), 2-15
 - (minus), 2-15
- active menugraph list, 8-6, 9-12
 - displaying, 8-9
- active MSC commands, current, listing, 6-21, 6-22
- ADD statement, 6-44
- adding new menu selections, 8-2
- ADM system command, B-3
- agenda (COMS), 8-13
 - MARCINPUT, 8-13
 - MARCOUTPUT, 8-13
- aging password, 2-2, 6-33
- ALEXIS internal form, 8-1
- ALLMSG command, 6-3
- <alternate family>, 6-6
- ALERTIME file attribute, 6-35
- ANYWINDOW directive option, 8-17
- ARRAYS task option, 4-9
- asterisk (*)
 - used to denote a super-user usercode, 7-3
 - used to enclose terms in online help, 3-3
- asynchronous job, 4-2
- attribute
 - dialogue, 5-4
 - file
 - ALERTIME, 6-35
 - displaying for files in list, 6-33
 - displaying values, 6-35
 - FILEKIND, 6-16, 6-32, 8-6
 - KIND, B-3
 - LOCKEDFILE, 6-44
 - MAXRECSIZE, 6-32
 - SECURITYTYPE of help book, 7-8
 - SECURITYTYPE of menugraph file, 8-5
 - sorting files in list by, 6-33
 - inherited, 4-1

Index

- job, 4-1
- print, 6-45
- security, 8-17
- task, 4-9, 4-10
 - assigning, 4-8
 - CONVENTION, 9-11
 - DESTNAME, 6-45
 - JOBSUMMARYTITLE, 6-47
 - LANGUAGE, 9-8
 - NOJOBSUMMARYIO, 6-47
 - PRINTDEFAULT, 6-48
 - SW1 through SW8, 4-8
- usercode
 - HOMEMENU, 8-3
 - MENUFILENAME, 8-7
- automatic display mode (ADM), B-3
- AUTORM task option, 4-9, 4-10
- AUTOSWITCHTOMARC task attribute, 4-16
- AX
 - action
 - on TASKSTATUS screen, 4-13, 4-14
 - selection
 - help text for, 3-1, 3-2
- AX (Accept) system command, A-1

- B**
- backup files, listing, 6-16
- Backup Processor utility, 6-16
- BACKUP task option, 4-9
- BASE task option, 4-9
- BD files, listing, 6-16
- BDBASE task option, 4-9
- BDIR command, 6-4, 6-16
- BDNAME task attribute, 4-10
- beginning-of-task (BOT) messages
 - displaying, 6-3
- billing, 1-3, 6-42, B-7
- billing support facility, A Series, 1-3, 6-42, B-7
- Billing Support library, 6-42
- binding a language
 - to MARC, 9-16
 - to the MCP, 9-5
- BMENU utility, 8-2
- BNA Version 1, 1-3
- BNA Version 2, 1-3
- BOOK/MARC/ENGLISH, 8-11, 9-13, B-2
- BOOK/MARC/SHORTHELP/ENGLISH, 8-11, 9-13, B-2

- book, help, 3-6, 7-8
 - default, 3-5
 - determining the language of, 9-12
 - generated by the Help Utility, 8-12
 - updating, 6-31
- BP command, 6-4
- break point, setting, 6-4
- BYE command, 2-18
- bypassing the menugraph search order, 8-11

- C**
- cache
 - help book, 6-31
 - screen, 6-31
- ccsversion, 9-3
 - accessing by application program, 9-4
 - setting, 9-3
- CENTRALSUPPORT library, 9-4
- CHANGE statement, 6-44
- changing the COMS application session
 - language, 6-10
- changing the MARC session language, 6-10
- character
 - control, 6-14
 - station control, B-6
 - string, 6-38
 - type, 9-2
 - wild-card, 6-19, 6-34
- character set
 - choosing, 9-3
 - coded, 9-2
 - EBCDIC, 9-1
 - LATIN1EBCDIC, 9-1
 - multiple, 9-1
- chargecode, 2-3, 7-6
 - screen, bypassing, 2-3
- Choice field, 2-6, 2-16
 - typeahead, 2-16
- CLASS option
 - SECOPT (Security Options) system
 - command, 2-4
- CLEAR control command, B-5
- CODE task option, 4-9
- coded character set, 9-2
- COL action, 2-16
- column range, in file display, 2-16
- Command and Edit (CANDE), window, 1-1
- command file, MSC, 6-18, 6-27
- command input screen, example of, 6-1
- command mode, 1-5

- command output screen, 2-17
- COMMANDCAPABLE status, 7-3
- commands, 6-1
 - available in COMS, C-2
 - available in MARC, 6-2, C-1
 - BYE, 2-18
 - CREATE STATION, 5-3
 - customizing, 8-14
 - displaying output from, 2-17
 - entering, 6-1
 - HELLO, 2-19
 - HELP, 3-5
 - ON, 5-2
 - primitive, 4-16, 6-43
 - RESUME, 5-3
 - screen-traversal, 8-2
 - SPLIT, 2-19
 - STATUS, 5-3
 - SUSPEND, 5-3
 - TO, 5-3
 - too long for Action field, 6-1
 - WRU, 5-3
 - ?CLOSE, 2-19
 - ?MARC, 4-16, 4-18
 - ?ON, 5-2
- commands, COMS, 6-42
 - alphabetical list of (table), C-2
 - entering, 5-1
- commands, data comm mode, B-4
- commands, MARC, 6-3
 - ALLMSG, 6-3
 - alphabetical list of (table), C-1
 - BDIR, 6-4, 6-16
 - BP, 6-4
 - DESTNAME, 6-45
 - DIRECTIVE, 8-14
 - FAMILY, 6-5
 - FILES, 6-7
 - GENERATE PASSWORD, 6-9
 - HELP, 8-12
 - JOBSUMMARY, 6-46
 - JOBSUMMARYTITLE, 6-47
 - LANGUAGE, 6-10, 8-8
 - LIST, 6-13, 6-16
 - MENU, 6-17, 8-9
 - MSC, 6-17
 - MSC, currently active, listing, 6-21, 6-22
 - MSG, 6-30
 - NEW, 6-31, 8-12
 - NEWS, 6-32
 - NOJOBSUMMARYTO, 6-47
 - PD, 6-7
 - PDIR, 6-33
 - PRINTDEFAULTS, 6-48
 - SHOW CCSVERSIONS, 9-3
 - SHOW LANGUAGES, 9-5
 - SHOW PASSWORD, 6-39
 - SMSG, 6-40
 - SUSPENDUSERCODE, 6-40
 - WRU, 6-41
 - ?CONTINUE, 6-5
- commands, primitive
 - alphabetical list of (table), C-10
 - ??MARC (Transfer to Menu-Assisted Resource Control), C-10
 - ??MEMDP (Dump Memory), 6-44, B-3, C-10
 - ??ODT (Return from Menu-Assisted Resource Control), 6-44, B-4
- commands, Print System, 6-45
 - alphabetical list of (table), C-10
- commands, system, 6-42
 - alphabetical list of (table), C-3
 - AX (Accept), A-1
 - DS (Discontinue)/inx>, 4-9, A-1
 - MSG (Display Messages), 6-30, 6-40, 6-43, 9-8, C-3
 - PD (Print Backup), 6-43
 - PD (Print Directory), 6-43
 - PP (Privileged Program), 7-6
 - REPLACE (Replace Disk or Pack Volume), 8-17
 - RESTART (Restart Jobs), 7-2
 - SECOPT (Security Options), 2-4, 7-7
 - CLASS, 2-4
 - MAXLOGONRETRIES option, 7-7
 - SL (Support Library), 6-29, 7-6, 8-11, B-2
 - SYSOPS (System Options), 9-2, 9-3, 9-5, 9-9, 9-10
 - TERM (Terminal), B-4
- common tasks, menu selections for, D-1
- Communications Management System (COMS), 5-1
 - commands, 6-42
 - alphabetical list of (table), C-2
 - entering, 5-1
 - setting security privileges with, 7-5
 - Utility, 7-2
- COMnd action, 2-14, 6-1
- COMND screen, example of, 6-1
- compile-time options, B-1
- computer system, A Series, 1-1

Index

COMS, (See Communications Management System (COMS))
COMS (Full-Featured), 7-5
 COMS Utility window and, 7-5
COMS (Kernel), 7-5
COMS application session language
 displaying or changing, 6-10
 setting, 9-7
COMS configuration file, 8-24
COMS Utility, 5-3, 7-2
COMS/ODT/DRIVER, B-4
configuration file, COMS, 8-24
configuring
 ODT, B-2
 user terminals, B-5
control character, 6-14
control-capable status, 7-2
CONVDF convention definition screen, 9-9
convention, 9-8
 choosing, 9-8
 customized, 9-9
 session, 9-10
 system, 9-9
convention changes
 using MARC screens, 9-9
CONVENTION task attribute, 9-11
conversation area
 of COMS message header, 8-13
COPY screen, example of, 2-11
COPY statement, 6-44
CP 2000 station, 6-53
 security considerations, 7-7
CREATE STATION command, 5-3
creating
 new forms, 8-3
 new menus, 8-2
CTRL key, 2-7
CU action
 on TASKSTATUS screen, 4-13
currently active MSC commands, listing,
 6-21, 6-22
cursor position
 when transmitting input, B-6
custom versions of MARC, creating, 8-1
customizing MARC, 8-1
 changing the language, 9-1
 forms, 8-3
 help text, 8-11, 8-12
 menugraphs, 8-1, 8-4
 usercode default home menu, 8-3
 menus, 8-2
 screens, 8-2

D

data comm
 mode, B-2
 terminal, 1-1
data comm terminal
 security considerations, 7-7
DBS task option, 4-9
DEBUG task option, 4-9
debugging
 and testing, 6-4
 utilities, 6-50
default
 ccsversion, setting, 9-3
 help books, 3-5
 message suppression control file, 6-29
 printer, designating, 6-48
 values, 2-12
 in input fields, 2-5
 overriding, B-7
 window, 5-3
definitions of terms, online, 3-3, 3-7
delay in recording of logon records, 2-4
deleting menu selections, 8-2
<destination>, 6-19
DESTNAME command, 6-45
DESTNAME task attribute, 4-10, 6-45
DIALOGINFO internal file, 8-33
dialogue, COMS, 1-1
 default settings for, 5-4
 defined, 5-1
 number permitted, 5-2
 switching, 5-2
directive
 format for, 8-19
 parameters of, 8-19
 result returned by, 8-24
 writing, 8-18
DIRECTIVE command, 8-14
 example of, 8-25
directives definition file, 8-18
directory
 displaying, 6-7
<directory name>, 6-7
display
 language, choosing, 9-4
 output, formatting, 8-13
displaying
 active menugraphs, 8-9
 command output, 2-17
 COMS application session language, 6-10
 contents of file, 6-13

- file attribute values, 6-35
- file attributes for files in list, 6-33
- keywords, 2-14, 3-6
- MARC session language, 6-10
- status of terminal, 6-41
- system option setting, 9-2
- DISPLAYONLYTOMCS task attribute, 4-20
- Documentation Library Overview, online, 3-7
- documents, A Series, viii, ix, x, xi
 - online, 1-5, 3-7
- DP action
 - on TASKSTATUS screen, 4-13
- DS (Discontinue) system command, 4-9, A-1
- DS action
 - on TASKSTATUS screen, 4-13
- DSED task option, 4-9
- dump, program, 4-8

E

- EBCDIC character set, 9-1
- Editor, A Series, 1-6
- embedded keyword in help text, 9-13
- end-of-task (EOT) messages
 - displaying, 6-3
- English language menugraph, 8-9
- English, system default language, 9-5, 9-6
- error messages, A-1
 - for NEW MENU command, 8-5
- error-handling procedures, A-1
- EXTERNALCOMMAND library procedure, 8-19

F

- FA action
 - on TASKSTATUS screen, 4-13
- FAMILY command, 6-5
- family specification
 - system default, 6-6
 - usercode default, 6-6
- <family specifications>, 6-6
- FAMILY statement, displaying, 4-6
- FAULT task option, 4-9
- fault-handling procedures, A-1
- field
 - Action, (See Action field)
 - Choice, (See Choice field)

- entering numbers instead of values in, 2-12
- input, 2-4, 2-12
- optional, 2-12
- overriding input in, B-6
- prefilled, B-7
- refreshing, 2-7
- sequence number, 6-16
- typeahead in Action field, 2-8
- file
 - access attempt, unauthorized, 2-4
 - attribute
 - ALERTIME, 6-35
 - displaying for files in list, 6-33
 - displaying values, 6-35
 - FILEKIND, 6-16, 6-32, 8-6
 - KIND, B-3
 - LOCKEDFILE, 6-44
 - MAXRECSIZE, 6-32
 - SECURITYTYPE, 8-5
 - sorting files in list by, 6-33
 - BOOK/MARC/ENGLISH, B-2
 - displaying information about, 6-8, 6-33
 - displaying name of, 6-7
 - listing contents of, 6-13
 - remote, 4-15
 - SUMLOG, 2-4
 - SYSTEM/<MCS name>, 8-18
 - SYSTEM/HELP, B-2
 - SYSTEM/MARC/AGENDA/TDXXX, 8-13, B-2
 - SYSTEM/MARC/COMMANDER, 9-14, B-2
 - SYSTEM/MARC/MENUGRAPH, B-2
 - SYSTEM/SUMLOG, 2-4
- file attribute
 - ALERTIME, 6-35
 - displaying for files in list, 6-33
 - displaying values, 6-35
 - FILEKIND, 6-16, 6-32, 8-6
 - KIND, B-3
 - MAXRECSIZE, 6-32
 - SECURITYTYPE, 8-5
 - sorting files in list by, 6-33
- file equation, 4-5, 4-7
 - defined, 4-7
- file name
 - help book, 8-11
 - help text source, 8-12
 - <file name>, 6-7, 6-14, 6-19, 6-47
 - <file title>, 6-13, 6-19, 6-47
- FILEEQUATE screen

Index

- example of, 4-7
- FILEKIND file attribute, 6-16, 6-32, 8-6
- FILES command, 6-7
- FILES task option, 4-9
- FIRST action, 2-15
- FM action
 - on TASKSTATUS screen, 4-13
- form, 1-1, 1-4, 2-11
 - command generation from, 2-13
 - customizing, 8-3
 - example of, 2-11
 - input fields, 2-12
 - internal, 8-1
 - localizing, 9-11
 - prompt, 2-12
- form field
 - typeahead, 2-8
- formatting display output, 8-13
- forms mode, B-5
- FR action
 - on TASKSTATUS screen, 4-14
- FROM qualifier, message ID, MSC command, 6-25
- full-screen transmission
 - configuring terminals for, B-6
- FUNCTIONS menu, 7-3, 7-4, 8-3

G

- GENERATE PASSWORD command, 6-9
- GETSTATUS procedure, 7-2
- Glossary, Master, online, 1-5, 3-7
- GO action, 2-14
 - bypassing the menugraph search order, 8-11
 - declaring a substitute menugraph, 8-10
 - restricting use of, 2-14, 7-4
- GO MARC command, 8-2
- guard file, 2-3

H

- halt/load
 - effect on active message suppression commands, 6-22
 - effect on MSC SUSPEND command, 6-21
 - effect on system ccsversion, 9-3
 - effect on system default language, 9-5

- effect on system messages stored for ALLMSG command, 6-3
- HELLO command, 2-19
- help, (See help text)
- help book, 1-6
 - security type, 7-8
 - source file names, 8-12, 9-13
 - updating, 6-31
- HELP command, 3-5, 8-12
- help information, redisplaying, 2-13
- HELP KEYS command, 3-6
- HELP REFERENCES command, 3-6
- help text, 3-2
 - book file name, 8-11
 - displaying, 3-2
 - example of, 3-2
 - generation of, 8-12
 - HELP command, 3-5
 - localizing, 9-12
 - on keywords, 3-5
 - screen, 3-2
 - TEACH command, 3-5
 - updating, 8-12
 - writing your own, 8-12
- HELP TOPICS command, 3-6
- Help Utility, A Series, 1-6, 8-12
- HELPSUPPORT library, 8-11
- HI action
 - on TASKSTATUS screen, 4-14
- Home action, 2-13
 - bypassing the menugraph search order, 8-11
- home menu, 2-10
 - usercode default, 8-3
- HOMEMENU usercode attribute, 8-3
 - of MAKEUSER utility, 8-3

I

- IB action
 - on TASKSTATUS screen, 4-14
- IL action
 - on TASKSTATUS screen, 4-14
- IMG, (See Interactive Menugraph Generator (IMG))
- improving MARC performance, 8-32
- in-use designation, menugraph or help book, 6-31
- InfoGuard, 2-4
- InfoGuard security enhancement, 7-1
- initializing MARC, B-7

initiating data comm mode at an ODT, B-2
input field, 2-4
 entering numbers instead of values in,
 2-12
 prefilled, B-7
input transmittal, 2-8, B-6
installing MARC, B-1
 files required for, B-1
Interactive Menugraph Generator (IMG),
 1-6, 8-1, 8-2, 9-13
internal form, 8-1

J

JC menu
 example of, 3-1
JDC menu
 example of, 2-10, 3-1
job, 4-1
 asynchronous, 4-2
 attribute
 inherited, 4-1
job number, 6-4
job summary
 information, creating file for, 6-47
 specifying printing of, 6-46
JOBSUMMARY command, 6-46
JOBSUMMARYTITLE, 6-47
JOBSUMMARYTITLE command, 6-47

K

key
 CTRL, 2-7
 selection, 1-1
 SPCFY, 1-4
 TAB, 2-2
 XMIT, 2-2, 2-8
KEys action, 2-14
KEYS keyword, 3-6
keyword
 displaying, 2-14, 3-6
 embedded, 9-13
 localizing, 9-13
KIND file attribute, B-3

L

language
 binding to MARC, 9-16
 binding to the MCP, 9-5
 bound to MARC, 6-10, 6-12, 9-4
 bound to the MCP, 9-4, 9-5, 9-6
 COMS application session, 6-10
 MARC session, 8-8
 setting
 COMS application session, 9-7
 MARC session, 9-6
 system, 9-5
 system, 8-8, 9-5, 9-6
 system default (English), 9-5, 9-6
LANGUAGE
 command, 6-10, 8-8
 set session language screen, 9-6
 task attribute, 4-10, 9-8
LAsT action, 2-16
LATIN1EBCDIC character set, 9-1
LIBRARIES task option, 4-9
library
 Billing Support, 6-42
 CENTRALSUPPORT, 9-4
 HELPSUPPORT, 8-11
 MARCSUPPORT, B-2
 NETEX, C-7
 of ALGOL procedures, 8-14
 PRINTSUPPORT, 6-49
 procedure
 using DIRECTIVE command, 8-25
 SECURITYSUPPORT, 2-2, 7-6, 7-7
 system, B-2
 SYSTEM/MARC/AGENDA/TDXXX, 8-13
list
 of active menugraphs
 displaying, 8-9
 of active MSC commands, 6-21, 6-22
LIST command, 6-13, 6-16
listing contents of file, 6-13
LOCAL menu, 9-3, 9-5, 9-9
local security procedures, writing, 2-2, 7-6
localization, 9-1
localizing
 action words, 9-12
 keywords, 9-13
 MARC messages, 9-14
 menus and forms, 9-11
 online documentation, 9-12
LOCKEDFILE file attribute, 6-44
log file

Index

- system, 6-24
- LOGANALYZER utility program, 6-24
- logging off, 2-18
- logging on, 2-1
 - exceptions, 2-3
 - when station becomes not ready, 2-2
- LOGOFFDISCONNECT, CP 2000 terminal
 - option, 6-53
- LOGOFFSTAT option, 6-52
- logon attempts, maximum, 7-7
- logon record, recording off, 2-4
- LOGON screen, 2-1
- long help text, 3-2
- LONG task option, 4-9

M

- Mail System, A Series, 4-6
- MAKEUSER utility, 8-7
 - HOMEMENU usercode attribute, 8-3
 - MENUFILENAME usercode attribute, 8-7
 - modifying the USERDATAFILE, 7-5
 - setting usercode default language, 9-7
 - specifying the home menu, 8-3
- MARC commands, 6-3
 - alphabetical list of (table), C-1
- MARC fault-handling procedures, A-1
- MARC home menu, 1-4, 2-10
- MARC menu, example of, 2-10
- MARC messages, localizing, 9-14
- MARC session language
 - displaying or changing, 6-10
 - menugraph, 6-10, 8-8
 - setting, 9-6
- MARCINPUT agenda, 8-13
- MARCOUTPUT agenda, 8-13
- MARCSUPPORT library, B-2
- Master Glossary, A Series, online, 1-5, 3-4, 3-7
- maximum
 - allowed logon attempts, 7-7
- MAXLINES task attribute, 4-10
- MAXLOGONRETRIES option
 - SECOPT (Security Options) system command, 7-7
- MAXRECSIZE file attribute, 6-32
- MCS, (See message control system (MCS))
- memory cache
 - for screens and help books, 6-31, 8-5
- menu, 1-1
 - command generation from, 2-17
 - customizing, 8-2
 - example of, 2-10
 - format, sample of two-column, 2-10
- FUNCTIONS, 7-4
- JC, 3-1
- JDC, 2-10, 3-1
- LOCAL, 9-3, 9-5, 9-9
- localizing, 9-11
- MARC home, 1-4, 2-10
- MSC message control, 6-22
- PS print system, 6-16, 6-45
- SC session control, 9-6, 9-10
- selections, 1-1, 1-4
 - adding, 8-2
 - deleting, 8-2
 - for common tasks, D-1
 - using, 2-10
 - with Micro A mouse, 2-11
- SSCONV show/set convention, 9-10
- SSLANG show/set language, 9-6
- station definition, COMS Utility, 5-3
- SYS system control, 6-22, 9-3, 9-5, 9-9
- TEACH, 3-7
- TOOLS, 9-14
- user, 7-6
- usercode default home, 8-3
- using, 2-10
- UTIL, 6-25

- MENU command, 6-17, 8-9
- menu mode, 1-4
- MENUFILENAME usercode attribute, 8-7
- menugraph, 1-6
 - active menugraph list, 8-6, 9-12
 - bypassing the search order of, 8-11
 - creating, 8-2
 - customizing, 8-1, 8-2
 - declaring a substitute, 8-10
 - default, 8-6
 - displaying active, 8-9
 - English language, 8-9
 - MARC session language, 8-8
 - navigating through, 2-9
 - search order of, 8-6
 - SECURITYTYPE, 8-5
 - software release level compatibility, 8-9
 - substitute, 8-7
 - system default, 8-8, 9-6
 - updating, 6-31, 8-4
 - upgrading, 8-2
 - usercode default, 8-7
- message control system (MCS), 5-1

message destination, MSC command, 6-26
 Message Handling System (MHS), 2-1
 message header, COMS
 conversation area, 8-13
 message ID, system, 6-19, 6-24
 warning, 6-19
 message suppression control file, default,
 6-29
 message suppression feature, 6-17
 Message Translation Utility (MSGTRANS)
 executing, 9-14
 used to bind a language to MARC, 6-10,
 6-11
 used to modify portions of MARC screens,
 8-3
 used to translate system messages, 9-8
 message type, MSC command, 6-23
 messages
 default display of, 6-17
 displaying, 5-3, 6-3, 6-30
 error, 8-5, A-1
 final task, 4-20
 in tasking, 4-19
 status, A-1
 storage, 5-3
 suppressed
 reactivating, 6-23
 suppressing display of, 6-17
 warning, 6-17, 6-19
 MESSAGESUPPORT function, 6-29
 MHS (Message Handling System), 2-1
 mix number, 4-12, 4-14, 6-3, 6-24, 6-25,
 6-40
 MLSMSGS array, establishing languages,
 9-14
 mode
 command, 1-5
 data comm, B-2
 commands, B-4
 forms, B-5
 menu, 1-4
 secure-video, 2-2
 single-line, B-5
 system command, B-2, B-3
 monitor station, 7-7
 monitoring task progress, 4-17
 mouse
 and help text, 3-8
 and menu selections, 2-11
 MSC
 command, 6-17
 currently active, listing, 6-21, 6-22

 command file, 6-18, 6-27
 message control menu, 6-22
 MSG (Display Messages) system command,
 6-30, 6-40, 6-43, 9-8, C-3
 MSG command, 5-3, 6-30
 MSGTRANS, (See Message Translation
 Utility (MSGTRANS))
 multihost network, 1-3
 multiple
 character sets, 9-1
 MARC dialogues, using, 5-2
 MARC menugraphs, using, 8-6

N

<name>, 8-15
 NETEX library, C-7
 network
 multihost, 1-3
 NEW command, 6-31
 NEW BOOK command, 6-31, 8-12
 NEW MENU command, 6-31
 updating the menugraph with, 8-4
 new menu
 creating, 8-2
 NEW MENU command, 8-4
 NEWS command, 6-32
 news file, 6-32
 NF action
 on TASKSTATUS screen, 4-14
 NOJOBSUMMARYIO
 command, 6-47
 task attribute, 6-47
 NOSUMMARY task option, 4-9
 not ready state, 2-2
 numbers
 entering instead of input field values in,
 2-12

O

ODT, (See operator display terminal (ODT))
 ODTs control command, B-5
 OF action
 on TASKSTATUS screen, 4-14
 OK action
 on TASKSTATUS screen, 4-14
 ON command, 5-2, 7-4
 one-time-only usercode, 2-2

Index

online
 documentation, (See help text)
 Documentation Library Overview, 3-7
 help, (See help text)
 term definitions, 1-5, 3-3, 3-4, 3-7
<op>, 6-34
Open Systems Interconnection (OSI), 1-3,
 2-1
operator display terminal (ODT), 1-2
 configuring, B-2
 running MARC at, B-5
 under control of COMS/ODT/DRIVER,
 B-4
optional fields, 2-12
OSI (Open Systems Interconnection), 1-3,
 2-1
OSI-MHS system, 2-1
OT action
 on TASKSTATUS screen, 4-14
OUTPUT screen
 example of, 2-17
overriding default delay in recording of logon
 records, 2-4

P

parameters
 used in EXTERNALCOMMAND library
 procedure, 8-19
PARENT action, 2-14
PASS command, 5-3
password, 2-2
 accesscode, 2-3
 aging, 2-2, 6-33
 changing, 7-6
 GENERATE PASSWORD command, 6-9
 one-time only, 2-2
 PASSWORD command, 6-33
 SHOW PASSWORD command, 6-39
 system-generated, 6-9
PASSWORD command, 6-33
password-generating system, 6-9, 6-39
pattern matching, 6-19
PB (Print Backup) system command, 6-43
PCMARC interface, 2-11
PD (Print Directory) system command, 6-43
PD command, 6-7
PDIR command, 6-33
performance of MARC, improving, 8-32
PP (Privileged Program) system command,
 7-6

PRev action, 2-13
 after using the LANGUAGE command,
 6-13
primitive commands, 4-16, 6-43
 ??MARC (Transfer to Menu-Assisted
 Resource Control), C-10
 ??MEMDP (Dump Memory), 6-44, B-3,
 C-10
 ??ODT (Return from MARC), 6-44
 ??ODT (Return from Menu-Assisted
 Resource Control), 6-44, B-4
primitive system commands
 alphabetical list of (table), C-10
print attribute, 6-45
Print System commands, 6-45
 alphabetical list of (table), C-10
PRINTDEFAULTS
 command, 6-48
 task attribute, 6-48
printer backup
 directory, displaying, 6-4
 files, listing, 6-16
PRINTSUPPORT library, 6-49
PRIVATELIBRARIES task option, 4-9
privileged status, 7-2
privileges, setting, 7-5
procedures
 writing directives, 8-18
process stack, 4-8
processing items, COMS, 8-13
program
 dump, 4-8
 output, tanked, 2-18
prompt, 2-12
PS commands, C-10, (See also Print System
 commands)
PS print system menu, 6-16, 6-45

Q

QT action
 on TASKSTATUS screen, 4-14
question mark (?), preceding commands, 5-1
QUIT control command, B-5

R

railroad diagrams, explanation of, E-1
reactivating suppressed messages, 6-23

- recording of logon records, 2-4
 - MARC default procedure, 2-4
 - overriding MARC default procedure, 2-4
 - redisplaying previous help information, 2-13
 - REFERENCES keyword, 3-6
 - refreshing the screen, 2-7
 - RELATIVE option
 - LIST command, 2-16
 - remote
 - file, 4-15
 - terminal, 6-17
 - REMOTE qualifier, MSC command, 6-23
 - REMOVE statement, 6-44
 - REM0001 remote window, 4-16
 - REPLACE (Replace Disk or Pack Volume)
 - system command, 8-17
 - RESTART (Restart Jobs) system command, 7-2
 - restricting the GO action, 2-14, 7-4
 - RESUME command, 5-3
 - REturn action, 2-15
 - RM action
 - on TASKSTATUS screen, 4-14
 - RUN
 - screen
 - example of, 4-5
 - running a task from, 4-5
 - selection, 4-1
 - statement, 6-44
 - running a task
 - assigning task attributes, 4-8
 - from the Action field, 4-10
 - making file equations, 4-7
 - specifying the code file name, 4-5
 - running MARC at an ODT, B-2
- S**
- SC session control menu, 9-6, 9-10
 - screen, 2-4
 - actions, (See actions)
 - cache, 6-31, 8-5
 - displaying from a previous menugraph, 8-11
 - CONVDF convention definition, 9-9
 - customizing, 8-2
 - LANGUAGE set session language screen, 9-6
 - layout, 2-5
 - localizing, 9-11
 - name, 2-6
 - prompt, 2-12
 - refreshing, 2-7
 - SC session control menu, 9-6
 - searching the menugraph for, 8-9
 - SSLANG show/set language menu, 9-6
 - SYSTEMCONVENTION change system convention, 9-9, 9-10
 - title, 2-6
 - types
 - command input, 6-1
 - command output, 2-17
 - forms, 2-11
 - help text, 3-2
 - home, 8-3
 - menus, 2-10
 - task status, 4-11
 - task view, 4-14
 - screen-traversal commands, 8-2
 - scrolling, 1-2
 - through output
 - backward, 2-15
 - forward, 2-15
 - search order, menugraph
 - active menugraph list, 8-6
 - bypassing, 8-11
 - displaying active menugraphs, 8-9
 - SECADMIN operating system option, 7-1, 7-2
 - SECOPT (Security Options) system
 - command, 2-4, 7-7
 - option
 - CLASS, 2-4
 - MAXLOGONRETRIES, 7-7
 - secure-video mode, 2-2
 - security, 7-1
 - attributes, naming conventions of, 8-17
 - categories, A Series, 7-1
 - effect on menu selections, 7-1
 - of help book, 7-8
 - privileges, setting, 7-5
 - setting privileges with COMS, 7-5
 - writing local procedures, 2-2, 7-6
 - SECURITY
 - command
 - for help book, 7-8
 - statement, 6-44
 - security-administrator status, 7-1
 - SECURITYSUPPORT library, 2-2, 7-6, 7-7
 - SECURITYTYPE file attribute
 - of help book, 7-8
 - of menugraph file, 8-5
 - selection description, 2-11

Index

- selection key, 1-1, 2-11
 - help, 3-5
 - selections for common tasks, D-1
 - SEQ action, 2-16
 - sequence number, 6-16
 - field, 6-16
 - in file display, 2-16
 - <sequence number control>, 6-14
 - serial numbers
 - in WFL statements, 6-44
 - session
 - convention
 - setting, 9-10
 - language
 - COMS application, 6-10
 - MARC, 8-8
 - level
 - changing system option setting, 9-2
 - messages, displaying, 6-30
 - number, 6-4
 - options
 - displaying, 9-2
 - setting, 9-2
 - print attributes, controlling, 6-45
 - session log-on, log-off, 2-4
 - SETSTATUS procedure, 7-2
 - setting
 - COMS application session language, 9-7
 - MARC session language, 9-6
 - session convention, 9-10
 - system convention, 9-9
 - short help text, 3-1
 - displaying, 3-1
 - example of, 3-1
 - SHOW CCSVERSIONS command, 9-3
 - SHOW LANGUAGES command, 9-5
 - SHOW option, MSC command, 6-23
 - SHOW PASSWORD command, 6-39
 - single-line mode, 2-3, B-5
 - SL (Support Library) system command,
 - 6-29, 7-6, 8-11, B-2
 - SLANG selection, SSLANG show/set
 - language menu, 9-6
 - SM system command, B-5
 - SMSG command, 6-40
 - SNA (Systems Network Architecture), 1-3
 - sorting
 - files in list by file attribute, 6-33
 - <source>, 6-19
 - SPCFY key, 1-4
 - SPLIT
 - action
 - on TASKSTATUS screen, 4-13
 - command, 2-19
 - SSCONV show/set convention menu, 9-10
 - SSLANG show/set language menu, 9-6
 - ST action
 - on TASKSTATUS screen, 4-14
 - stack, process, 4-8
 - standard conventions, 9-9
 - START
 - screen
 - starting a WFL job from, 4-4
 - selection, 4-1
 - statement, 6-44
 - statements
 - Work Flow Language (WFL)
 - entered as commands, 6-44
 - statements entered as commands, WFL
 - alphabetical list of (table), C-12
 - station
 - CP 2000
 - security considerations, 7-7
 - station control character, B-6
 - station definition menu, COMS Utility, 5-3
 - station status, displaying, 6-41
 - STATION task attribute, 4-10
 - station, monitor, 7-7
 - status
 - line, 2-7
 - messages, A-1
 - of terminal, displaying, 6-41
 - STATUS command, 5-3, 6-52, (See also WRU command)
 - string, character, 6-38
 - <substitute family>, 6-6
 - substitute menugraph, 8-6
 - declaring, 8-10
 - SUMLOG file, 2-4
 - super-user status, 7-3
 - super-user-capable station, 2-2, 7-3
 - defined, 7-5
 - SUPPRESS option, MSC command, 6-23
 - suppressed messages, reactivating, 6-23
 - suppressing system messages, 6-17
 - SUSPEND command, 5-3
 - SUSPENDUSERCODE command, 6-40
 - switching windows, 5-2
 - SW1 through SW8 task attributes, 4-8
 - SW1 to SW8 system switches, 4-8
 - SYMBOL/MARC/ENGLISH, 8-12
 - SYMBOL/MARC/SHORTHELP/ENGLISH,
 - 8-12
- synchronous task, 4-2

- SYS system control menu, 6-22, 9-3, 9-5, 9-9
- SYSOPS (System Options) system command, 9-2, 9-3, 9-5, 9-9, 9-10
- system
- command, 6-42
 - ADM, B-3
 - alphabetical list of (table), C-3
 - AX (Accept), A-1
 - DS (Discontinue), 4-9, A-1
 - mode, B-2, B-4
 - MSG (Display Messages), 6-30, 6-40, 6-43, 9-8, C-3
 - PB (Print Backup), 6-43
 - PD (Print Directory), 6-43
 - PP (Privileged Program), 7-6
 - REPLACE (Replace Disk or Pack Volume), 8-17
 - RESTART (Restart Jobs), 7-2
 - SECOPT (Security Options), 2-4, 7-7
 - SL (Support Library), 6-29, 7-6, 8-11, B-2
 - SM, B-5
 - SYSOPS (System Options), 9-2, 9-3, 9-5, 9-9, 9-10
 - TERM (Terminal), B-4
 - command mode, B-3
 - control terminal, 1-2
 - convention
 - setting, 9-9
 - default
 - family specification, 6-6
 - language (English), 9-5, 9-6
 - menugraph, 8-8, 9-6
 - language
 - defined, 8-8
 - setting, 9-5
 - level
 - changing system option setting, 9-2
 - setting ccsversion, 9-3
 - log file, 2-4, 6-24
 - messages
 - default display of, 6-17
 - displaying, 6-3, 6-40
 - ID, 6-19, 6-24
 - suppressing display of, 6-17
 - option setting
 - changing at session level, 9-2
 - changing at system level, 9-2
 - displaying, 9-2
 - options
 - displaying, 9-2
 - setting, 9-2
 - switches SW1 to SW8, 4-8
 - utilities, initiating, 6-49
 - warning messages, 6-19
 - system-generated password, 6-9, 6-39
 - SYSTEM/<MCS name> file, 8-18
 - SYSTEM/HELP file, B-2
 - as a system library, B-2
 - SYSTEM/LOGANALYZER utility program, 6-24
 - SYSTEM/MARC/AGENDA/TDXXX file, B-2
 - SYSTEM/MARC/AGENDA/TDXXX library file, 8-13
 - SYSTEM/MARC/COMMANDER file
 - adding localized language, 9-14, B-2
 - installing, B-2
 - SYSTEM/MARC/MENUGRAPH file, B-2
 - SYSTEM/SUMLOG, 2-4
 - SYSTEMCONVENTION change system
 - convention screen, 9-9, 9-10
 - Systems Network Architecture (SNA), 1-3
 - SYSTEMUSER status, 7-2
- T**
- TAB key, 2-2
 - TADS, (See Test and Debug System (TADS))
 - TADS task attribute, 4-10
 - tanked program output, 2-18
 - TANKFILE internal file, 8-33
 - <target family>, 6-6
 - task, 4-1
 - attribute
 - assigning, 4-8, 4-10
 - AUTOSWITCHTOMARC, 4-16
 - BDNAME, 4-10
 - CONVENTION, 9-11
 - DESTNAME, 4-10, 6-45
 - DISPLAYONLYTOMCS, 4-20
 - inherited, 4-1
 - JOBSUMMARYTITLE, 6-47
 - LANGUAGE, 4-10, 9-8
 - MAXLINES, 4-10
 - NOJOBSUMMARYIO, 6-47
 - option, 4-9
 - PRINTDEFAULTS, 6-48
 - STATION, 4-10
 - TADS, 4-10
 - beginning and ending of, recorded in system log file, 2-4

Index

- common
 - menu selections for, D-1
 - initiating, 4-17
 - messages
 - displaying, 6-30
 - option
 - ARRAYS, 4-9
 - AUTORM, 4-9, 4-10
 - BACKUP, 4-9
 - BASE, 4-9
 - BDBASE, 4-9
 - CODE, 4-9
 - DBS, 4-9
 - DEBUG, 4-9
 - DSED, 4-9
 - FAULT, 4-9
 - FILES, 4-9
 - LIBRARIES, 4-9
 - LONG, 4-9
 - NOSUMMARY, 4-9
 - PRIVATELIBRARIES, 4-9
 - TODISK, 4-9
 - TOPRINT, 4-9
 - progress, monitoring, 4-17
 - running, 4-5
 - suspending, 4-17, 4-19
 - synchronous, 4-2
 - terminating, 4-19
 - window, 4-15
- TASK action, 4-18
 - on TASKSTATUS screen, 4-13
- TASKATTR screen
 - example of, 4-8
- tasking mode
 - defined, 4-17
 - entering, 4-17, 6-44
 - exiting, 4-18
 - final message, 4-20
- TASKSTATUS screen, 4-11
 - actions on, 4-13
 - example of, 4-11
- TASKVIEW screen, 4-14
 - example of, 4-14
- TCP/IP (Transmission Control Protocol/Internet Protocol), 1-3
- TEACH
 - command, (See HELP command)
 - menu, 3-7
- Teletype interaction, B-5
- TERM (Terminal) system command, B-4
- term definitions, online, 1-5, 3-3, 3-4, 3-7
- terminal
 - configuring, B-5
 - CP 2000
 - option, LOGOFFDISCONNECT, 6-53
 - data comm
 - security considerations, 7-7
 - operator display (ODT), 1-2
 - remote, 6-17
 - status, displaying, 6-41
 - system control, 1-2
 - types, B-5
 - user, 1-2- TERMINAL NAME attribute, B-6
- terms, A Series, online, 1-5, 3-4, 3-7
- Test and Debug System (TADS), 6-4
- TI action
 - on TASKSTATUS screen, 4-14
- time of day, 2-6
 - updating, 2-7
- TO command, 5-3
- TODISK task option, 4-9
- TOOLS menu, 9-14
- TOPICS keyword, 3-6
- TOPRINT task option, 4-9
- Transmission Control Protocol/Internet Protocol (TCP/IP), 1-3
- transmitting input, 2-8, B-6
- tutorial, online, 3-7
- type, character, 9-2
- <type>, 6-18
- typeahead
 - in Action field, 2-8
 - in Choice field, 2-16

U

 - UL action
 - on TASKSTATUS screen, 4-14
 - unauthorized file access attempt, 2-4
 - updating
 - help book, 6-31, 8-12
 - menugraph, 6-31, 8-4
 - time of day display, 2-7
 - upgrading customized MARC menugraphs, 8-2
 - USER menu, 7-6
 - user terminal, 1-2
 - user-defined commands in MARC, 8-14
 - user, A Series, 1-1
 - usercode, 2-2
 - attribute
 - HOMEMENU, 8-3

MENUFILENAME, 8-7
 COMMANDCAPABLE, 7-3
 control capable, 7-2
 default family specification, 6-6
 default home menu, 8-3
 default menugraph, 8-7
 one-time-only, 2-2
 password changes, 7-6
 privileged, 7-2
 security administrator, 7-1
 suspending, 6-40
 SYSTEMUSER, 7-2
 USERDATAFILE usercode database, 8-7
 setting security privileges with, 7-5
 usercode privileges, 7-5
 UTIL menu, 6-25
 utilities, initiating through MARC, 6-49
 debugging, 6-50
 general, 6-50
 system administrator, 6-49

V

VIEW action, 4-18
 on TASKSTATUS screen, 4-13
 view screen, 4-14
 actions on, 4-15
 example of, 4-14

W

warning message, 6-17, 6-19
 WFL, (See Work Flow Language (WFL))
 wild-card character, 6-19, 6-34
 window (COMS), 5-1
 default, 5-3
 switching, 5-2
 window, task, 4-15
 Work Flow Language (WFL)
 job
 from the Action field, 4-4
 from the START screen, 4-3
 programs
 running from MARC, 4-1
 statements
 alphabetical list of (table), C-12
 entered as commands, 6-44, C-12
 serial numbers in, 6-44
 WRU command, 5-3, 6-41, 6-52

X

X.25, 1-3
 XMIT key, 2-2, 2-8

Y

Y action
 on TASKSTATUS screen, 4-14

.KEYWORD command, 9-13
 + action, 2-15
 * usercode, 7-3
 *MESSAGES/STARTUP, 6-29
 - action, 2-15
 ??DP (Dump) primitive command, 6-44
 ??MARC (Transfer to MARC) primitive
 command, B-4
 ??MARC (Transfer to MARC) primitive
 command, C-10
 ??MEMDP (Dump Memory) primitive
 command, 6-44, B-3, C-10
 ??ODT (Return from MARC) primitive
 command, 6-44, B-4
 ?CLOSE command, 2-19
 ?CONTINUE command, 6-5
 ?MARC command, 4-16, 4-18
 ?ON command, 5-2
 ?PASS command, 5-3
 ?WRU command, 6-41

Help Us To Help You

Publication Title _____

Document 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

Recommended Change (Please identify affected pages) _____

Name _____

Telephone Number _____

Title _____

Company _____

Address (Street, City, State, Zip) _____

Help Us To Help You

Publication Title _____

Document 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

Recommended Change (Please identify affected pages) _____

Name _____

Telephone Number _____

Title _____

Company _____

Address (Street, City, State, Zip) _____

Help Us To Help You

Publication Title _____

Document 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

Recommended Change (Please identify affected pages) _____

Name _____

Telephone Number _____

Title _____

Company _____

Address (Street, City, State, Zip) _____



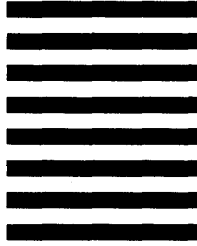
No Postage
necessary
if mailed in the
United States

BUSINESS REPLY MAIL

First Class Permit No. 817 Detroit, MI 48232

Postage Will Be Paid By Addressee

Unisys Corporation
ATTN: Publications
25725 Jeronimo Road
Mission Viejo, CA 92691-9826 USA



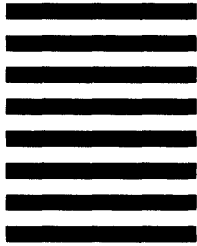
No Postage
necessary
if mailed in the
United States

BUSINESS REPLY MAIL

First Class Permit No. 817 Detroit, MI 48232

Postage Will Be Paid By Addressee

Unisys Corporation
ATTN: Publications
25725 Jeronimo Road
Mission Viejo, CA 92691-9826 USA



No Postage
necessary
if mailed in the
United States

BUSINESS REPLY MAIL

First Class Permit No. 817 Detroit, MI 48232

Postage Will Be Paid By Addressee

Unisys Corporation
ATTN: Publications
25725 Jeronimo Road
Mission Viejo, CA 92691-9826 USA





86000403-100